LiNUX+DVD Nº 61

LiNUX+

LA MAYOR REVISTA ONLINE SOBRE LINUX

Nº 1/2010 (61) MENSUAL ISSN 1732-7121

SDL
INTERFACES GRÁFICAS DE USUARIO

WEKA
PROGRAMACIÓN INTELIGENTE

389 DIRECTORY SERVER
ALTERNATIVA LIBRE AL ACTIVE DIRECTORY DE MICROSOFT

LINUX EN EL INSTITUTO
IMPLEMENTACIÓN DE UN SERVIDOR

THEREMÍN VIRTUAL
UN INSTRUMENTO MUSICAL DE NUEVA GENERACIÓN

SERVIDORES DNS
DOMINIOS, SUBDOMINIOS Y ZONAS

CÓMPUTO FORENSE
HERRAMIENTAS PARA LA ADQUISICIÓN DE DATOS

MAKE
COMPILACIÓN INTELIGENTE



A TODO AQUEL QUE CREE QUE ERES UN FREAK, DEMUÉSTRALE QUE TIENE RAZÓN.
PREMIAMOS A LAS 10 MEJORES IDEAS DE APLICACIONES MAEMO.
¿Qué aplicaciones te gustaría tener en tu móvil? Ahora, con el nuevo Nokia N900 y la plataforma Maemo. Cuéntanos tus ideas.
Las 10 mejores tienen premio. Entra en nokia.es/n900
NOKIA
Nseries
NOKIA
N900
12:40
ovi
Maemo software | No
Nokia España
Madrid
Sevilla
Algiers
Rabat
MOROCCO
ALGERIA
Introducción

Linux+, ¡Por fin liberado!

¡Hola amigos y amigas! Me alegro muchísimo de poder presentaros el primer número de Linux+ en versión on-line totalmente libre. Aunque para muchos de vosotros será la misma revista, aunque en otro formato, creo que con este primer número empezamos algo nuevo. Al empezar el trabajo en Linux+ hace cuatro años, ni podía imaginar que un día podríamos distribuir la revista entre toda la gente interesada, independientemente de su nacionalidad y lugar de residencia. Publicando la revista en versión papel nos vimos limitados a ciertos puntos de venta y un número de ejemplares concreto. Muchas veces nos escribía gente preguntando dónde se podía comprar la revista porque no la había en los kioscos cercanos. Y en los grandes puntos de venta donde sí se podía comprar, a menudo desaparecía muy rápido.

Por otra parte recibimos muchos emails de las personas que argumentaban que una revista sobre Linux debería ser libre como este sistema operativo. Pero claro, es imposible con el coste de impresión, transporte, distribución, etc. Con el formato electrónico ya es un poco más viable.

Algunos de vosotros dirán que no es lo mismo la versión electrónica que la impresa en papel. Es verdad, a mí también me gusta mucho hojear la revista, incluso olerla es agradable, pero seamos sinceros, en este mundo no se puede tener todo.

Espero que las ventajas de recibir la revista en casa el primer día de cada mes (excepto este número que os enviamos antes como un regalo de Navidad), poder descargarla e incluso compartir con los amigos, sin ningún coste, sean suficientes para que veáis este cambio con buenos ojos.

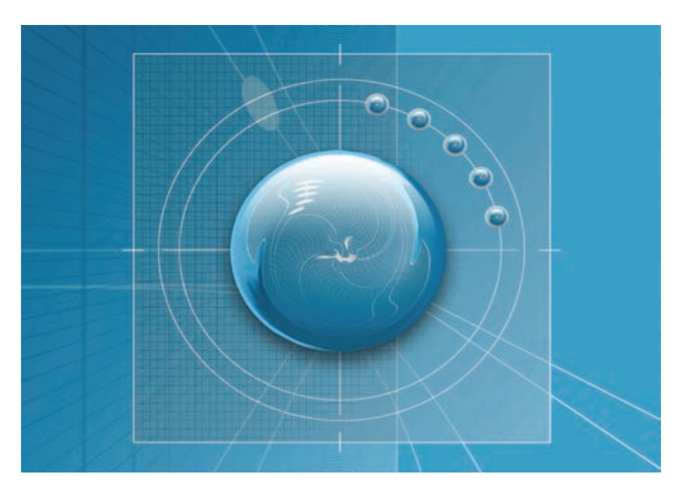
Con mucho gusto recibiremos vuestras opiniones (tanto positivas como negativas) acerca de nuestra decisión; las opiniones más interesantes y bien argumentadas las publicaremos en el número de febrero. Todos los que quieran ayudarnos compartiendo su punto de vista, por favor que escriban a: `es@lpmagazine.org` con el tema Cambios en Linux+.

Y para terminar, como la encuesta en nuestra web muestra que Programación es uno de los tres temas más elegidos por vosotros (además de Seguridad y Hacking), le dedicamos este número. Espero que los materiales incluidos os resulten novedosos y prácticos y que disfrutéis mucho de esta nueva versión de vuestro viejo amigo Linux+.

¡Ah, casi me olvidaba, todo lo mejor para el año 2010!

Paulina Pyrowicz
Redactora Jefe de Linux+

En este número





Open-SLX vende OpenSUSE 11.2 en caja

Esta noticia me trae gratos recuerdos de cuando las cajas oficiales de SuSE podían comprarse fácilmente en nuestro país (que de hecho era la única manera de tener la versión completa de esa distribución). Personalmente llegué a poseer varias y realmente era muy conveniente en aquellos días de Internet dial-up por su generosa documentación impresa y múltiples CDs.Ahora un anterior empleado de SUSE fundó Open-SLX, un emprendimiento aprobado por Novell que ofrece una versión "comercial" y en caja del proyecto comunitario OpenSUSE. Su nueva versión, disponible desde hace sólo una semana, incluye a OpenSUSE 11.2 con su propio soporte técnico para la instalación y ya se vende en Europa por € 59,95; próximamente también en Norteamérica. En comparación SUSE Linux Enterprise Desktop tiene un precio estándar de subscripción de U$S 120 por dispositivo por año. Para el próximo lanzamiento de OpenSUSE 11.3, Open-SLX planea agregar más valor a su paquete y diferenciarse más del proyecto comunitario ofreciendo una "diferencia real" con la versión de OpenSUSE que se puede descargar gratuitamente.
http://www.vivalinux.com.ar/distros/open-slx-11.2

VirtualBox 3.1 ahora con teleportación

Sun anunció el lanzamiento de la versión definitiva de VirtualBox 3.1, la nueva gran actualización de su popular software de virtualización para arquitecturas x86, que ahora incluye grandes novedades como por ejemplo la "teleportación" también conocida como migración en vivo, que permite migrar máquinas virtuales mientras se están ejecutando entre distintos anfitriones, aún si estos tienen CPUs distintos (como Intel o AMD). La "teleportación" debería ser especialmente apreciada por los administradores de sistemas, que ahora podrían efectuar tareas de mantenimiento en sus máquinas virtuales sin ninguna anomalía o interrupción en los servicios que ofrecen. Esta nueva característica de VirtualBox lo hará más competitivo en una serie de mercados en los que anteriormente no era tan popular, como los Datacenters. Otras novedades incluyen aceleración de vídeo 2D para huéspedes Windows, soporte para más de una unidad de almacenamiento óptica permitiendo que pueda haber más de una unidad conectada por cada controlador, restauración de "instantáneas" (snapshots) en orden arbitrario y la posibilidad de tomar instantáneas de otras instantáneas (branching), interfaces de red paravirtualizadas con una configuración que puede cambiarse sin detener la máquina virtual, mejoras significativas en el desempeño en huéspedes AMD64. Además han resuelto un montón de problemas. Puedes ver el registro de cambios y novedades en
http://www.vivalinux.com.ar/soft/virtualbox-3.1

Google lanzará su propio teléfono en 2010

Muchos han sido los rumores en los últimos días sobre el lanzamiento de Google de su propio terminal y la información parece haber cobrado fuerza. Ha sido la propia compañía, a través de una entrada en su blog corporativo, quien ha confirmado la fabricación de un teléfono en colaboración con un socio no desvelado, el cual ya estaría disponible para ser testeado por los propios trabajadores de Mountain View.

El terminal, denominado "Nexus One" se venderá libre y sin estar asociado a ningún operador, según fuentes cercanas a la compañía. Se tratará de un smartphone que incorporará el sistema operativo Android 2.1 y podría tener una pantalla OLED capacitiva similar a la del HTC Dragon, sin teclado físico. Incluiría además el procesador Snapdragon y pantalla táctil OLED, dos micros y especiales capacidades de transformación de voz en texto.

Según afirma The Wall Street Journal, el dispositivo se pondrá a la venta el próximo año y Google lo comercializará directamente. Serán los usuarios quienes decidan con quien contratar sus servicios inalámbricos. Esto proporciona a la compañía una gran libertad para incluir software de servicios como su e-mail o su herramienta de mapas sin estar sujeto a las normas de las operadoras.

El lanzamiento deja entrever las intenciones de la firma de competir directamente con el iPhone de Apple y no deja de ser curioso



Se llamará Nexus One y ya ha empezado a ser probado por los trabajadores de la firma de Mountain View

que pueda presentarse sin depender de ninguna otra empresa. Últimamente, entre operadoras y fabricantes de teléfonos, un total de 32 empresas han anunciado el lanzamiento de algún teléfono con el sistema Android.

Las primeras imágenes muestran un diseño similar al del iPhone con un único botón en la parte inferior, debajo de la pantalla, que actúa en realidad como joypad.

De momento se desconocen los precios de este dispositivo así como los países donde podría desembarcar. Según afirmaciones, se comercializará a partir de enero del año que viene.

http://www.itespresso.es/es/news/2009/12/14/google-phone-2010

Virtualización en Linux reducirá sus requerimientos de memoria

La próxima versión del kernel de Linux, que está pronta a ser liberada oficialmente (2.6.32), incluirá un cambio que beneficiará a todos los que estén usando virtualización sobre Linux, ya que se implementó una técnica para reducir drásticamente el uso de memoria. La virtualización es una técnica para compartir un mismo hardware sobre varios sistemas operativos en forma simultánea. Cada sistema operativo corre en forma independiente, sin enterarse de que está compartiendo el hardware. De esta forma se pueden combinar varios servidores o máquinas en una sola, ideal para casos en donde no siempre están todas ocupadas, ya que el mismo hardware no ocupado por una máquina virtual está disponible para otra.

Aunque Linux en general reutiliza la memoria cuando dos o más aplicaciones usan la misma biblioteca, Kernel Sampage Merging (KSM) es una nueva característica de Linux que lleva la reutilización del hardware un paso más allá: en sistemas virtualizados es común que estén en ejecución varias copias de un mismo sistema operativo e incluso varias copias de una misma aplicación o servicio, con KSM todas estas copias se detectan y consolidan en una sola, reduciendo drásticamente la cantidad de memoria que se necesita. Por ejemplo Red Hat indica que en sus pruebas ha tenido funcionando 600 máquinas virtuales en un solo host de 48 cores y 256 GB de RAM sin ningún tipo de problemas.



Implementación de KSM es rápida, limpia y transparente. A nivel conceptual la técnica es sencilla, y se trata de lo siguiente: un sistema operativo divide la memoria en múltiples páginas de tamaño fijo, gracias a esta división se pueden guardar las páginas de memoria no ocupadas en disco y posteriormente recuperarlas llevándolas a memoria cuando se necesitan (swapping), con esto se puede usar más memoria de la disponible físicamente. KSM calcula un identificador único por cada página en base a los datos que contiene (hash) y si detecta que tiene los mismos datos que otra página (mismo hash) entonces descarta la duplicada dejando sólo una, y hace pensar al resto del sistema que se trata de páginas diferentes, pero físicamente es una sola. Si una aplicación modifica la página, entonces se crea un duplicado y se obtiene una página propia, como si siempre hubiesen estado separadas.

Esto quiere decir que si tenemos dos máquinas virtuales que ocupan 768 MB pero de esa memoria 512 MB son idénticos porque se trata del mismo sistema operativo + bibliotecas + aplicaciones, entonces en vez de usar 1,5 GB sólo se requiere 1 GB.

KSM también se ha implementado en versiones anteriores del kernel (backporting) y ya se encuentra disponible en Fedora12. Aunque es en virtualización en donde se obtienen los mayores beneficios, los usuarios de Fedora pueden ver si le están sacando provecho revisando el archivo virtual /sys/kernel/mm/KSM/pages_sharing para ver cuantas páginas físicas se están compartiendo entre una o más páginas lógicas.

http://www.fayerwayer.com/2009/12/virtualizacion-en-linux-reducira-sus-requerimientos-de-memoria/

FSF trabaja con Paypal en beneficio de la comunidad del software libre

Un gran número de personas en la comunidad del software libre sienten que PayPal es una manera conveniente de enviar dinero a otros. Parte de la razón de esto es que puedes utilizar muchos de los servicios de PayPal, sólo con software libre -ya que normalmente no requieren un software propietario especial, ni siquiera JavaScript.

Sin embargo, la FSF descubrió recientemente que PayPal añadió una licencia de software propietario en su Acuerdo de Usuario. El ingeniero Brett Smith en conformidad con la licencia de la FSF explicó: "Por supuesto que, la FSF no está de acuerdo con estos términos, así que tan pronto nos enteramos de ello, nos contactamos con PayPal, para ver si podíamos hacer otro acuerdo. La compañía escuchó nuestra preocupación, y comprendió nuestra posición frente a estas condiciones. Y no sólo eso: el próximo año, PayPal también actualizará su acuerdo de usuario para asegurarse que la comunidad del software libre puede continuar recibiendo y haciendo pagos sin tener que aceptar una licencia de software propietario".

El director ejecutivo de la FSF, Peter Brown dijo: "Nos gustaría expresar nuestro agradecimiento a PayPal por tomarse el tiempo para escucharnos y hacer este cambio."

http://www.linuxtoday.com/developer/2009112402535PRLL

Creador de MySQL hace llamado para salvarlo de Oracle

Definitivamente la adquisición de Sun por parte de Oracle sigue generando un importante rechazo al interior del grupo a cargo de uno de los proyectos más emblemáticos de Sun: MySQL.

Si bien el Departamento de Justicia de Estados Unidos ya dio el visto bueno para la adquisición, desde la Comunidad Europea se han mostrado bastante más contrarios a esta idea. Tal vez es esta la razón que motivó a Michael "Monty" Widenius (creador de MySQL) a realizar un llamado para que seamos todos quienes demostremos nuestro descontento con la idea de que Oracle tome el control de MySQL.

Quienes deseen participar de esta campaña deben enviar un correo electrónico a la dirección comp-merger-registry@ec.europa.eu, adjuntando sus datos personales y una descripción de las razones por las cuales te verías afectado si Oracle finalmente logra quedarse con MySQL.

http://monty-says.blogspot.com/2009/12/help-saving-mysql.html

KOffice 2.1, en el camino correcto

KOffice es una suite ofimática multiplataforma, libre y de código abierto para el proyecto KDE. El equipo de desarrolladores de KOffice acaba de anunciar la publicación de KOffice 2.1, una versión que aparece 6 meses después del lanzamiento de la versión 2.0 y que incluye "una serie de nuevas características y además mejoras generales en la madurez de aplicaciones individuales".

Destacan la importación de documentos, y según el equipo parece que el código base, que está mejor estructurado desde la versión 2.0, empieza a dar sus frutos, con muchas mejoras a pesar del limitado grupo de desarrolladores. La versatilidad de esta suite ofimática se confirma con cosas como el anuncio de que se usará en los Nokia N900 basados en Maemo Linux.

Esta versión es una mejora notable de casi todas las partes de KOffice comparado con la versión anterior. KOffice 2.0 fue calificada de "plataforma de lanzamiento", lo que se entiende como un primer adelanto del marco de trabajo y el nuevo paradigma en la interfaz de usuario. En la versión 2.1, la mayoría de las aplicaciones y componentes han mejorado significativamente, pero aún no se encuentra en un nivel para ser consideradas como las herramientas principales de trabajo. Las excepciones son las aplicaciones de gráficos: Krita, el editor de imágenes, y Karbon, el editor de graficos vectoriales.

Todas las aplicaciones de KOffice tienen un nuevo diseño en la interfaz gráfica de usuario y están mejor adaptadas a las pantallas anchas actuales. La interfaz gráfica de usuario se compone de un área de trabajo y una barra lateral donde las herramientas se encuentran acopladas. Cualquier herramienta puede ser arrastrada para crear su propia ventana y más tarde ser re-acoplada, otorgando una total flexibilidad. La ubicación de las herramientas, es guardada y reutilizada en las próximas sesiones que se utilice KOffice.

http://www.koffice.org/news/koffice-2-1-released/



Ubuntu B-Sides

Si te quejas que todavía Canonical no incluye en la instalación por defecto de Ubuntu gran cantidad de software que crees imprescindible, la solución se llama Ubuntu B-Sides (de las famosas caras B del mundo de la música) que incluye gran cantidad de software extra que aún no se incluye, pero que por popularidad y funcionalidad, se incluirá en breve en la distribución oficial.
Si tienes ya instalado Ubuntu 9.10 y quieres instalar las caras B haz lo siguiente en la consola:

- Añade el repositorio:

```
sudo add-apt-repository
ppa:b-sides/ppa
```

- Actualiza la lista:

```
sudo apt-get update
```

- Instala Ubuntu B-Sides:

```
sudo aptitude install b-sides
```

Dell vuelve a ofrecer equipos con Ubuntu preinstalado

Hace unos meses, de manera inesperada, Dell (la que fue en su día la primera empresa en confiar en Ubuntu para instalarlo en sus equipos) decidió que no vendería más equipos con Ubuntu preinstalado.
La verdad que la noticia sentó como un "jarro de agua fría" entre los usuarios de Linux, ya que parecía que este tipo de pasos no tenían marcha atrás. Las razones esgrimidas por la empresa eran de carácter económico. Pasado el tiempo, Dell ha dado marcha atrás en su decisión y ha decidido volver a vender equipos con Ubuntu preinstalado. A priori, los equipos traen Ubuntu 9.04 (la compatibilidad ha de ser total y por eso todavía, en el momento de escribir este artículo no "montaban" la Karmic Koala), pero en breve esperamos que se actualicen a la nueva versión.
Para terminar, contaros que hay diferentes ordenadores que brindan esta posibilidad: Inspiron 537 n-Series ST, netbooks Dell Mini v10 (con la versión 9.04 o Ubuntu Netbook Remix), 15n Inspiron y Studio XPS 13.

Canonical limita el envío de CD vía ShipIt

El programa que tantos CD de Ubuntu ha enviado a todos los lugares del mundo de forma gratuita, ShipIt, está sufriendo por primera vez en su historia fuertes limitaciones. Aunque hace unos años limitaron el envío a particulares de más de un CD de forma masiva (aún recuerdo cuando podías pedir hasta 10 creo que era), este año Canonical ha restringido más aún su política de envíos: los que hayan pedido un CD un año anterior no podrán volver a pedirlo. La idea es limitar los enormes costes que a la compañía le supone el envío de CD de una forma tan masiva. Los organismos o asociaciones, podrán seguir pidiendo los CD de igual manera y los que nunca han probado el programa, podrán probarlo y probablemente tengan el CD mucho antes que cuando el envío era tan masivo.

Mirando con perspectiva Karmic Koala

Hace ya bastante tiempo que salió a la red la última versión de Ubuntu y podemos mirar con perspectiva qué mejoras ha traído, y qué mejoras deberían traer las futuras versiones. A pesar de los problemas iniciales, con el tiempo, todo se resolvió y parece que la nueva versión es más sólida que las precedentes.

En un primer momento, los portales llegaban a hablar de cifras de hasta un 85% de instalaciones con algún defecto. Yo mismo sufrí problemas, pero he de reconocer que se trataba de la versión *Release Candidate* así que es entendible ya que se trata de una versión para pruebas (en la versión para *netbooks*, la UNR que instalé en mi Eee he de reconocer que no he tenido ningún problema desde el primer día). Con el tiempo me decidí a instalar la versión nueva, la estable, y no tuve ningún problema en un comienzo.

Sin embargo, aunque el número de usuarios sigue creciendo y cada vez es algo más conocido, a Ubuntu le queda un largo camino por recorrer. Los hechos demuestran que el mejor no siempre gana, y menos aún en este mercado, pero todavía quedan cosas por mejorar.

Desde mi humilde punto de vista, algunas de las cosas que mejoraría en Ubuntu para atraer a un público más variado puede resumirse en los siguientes puntos.

Mejora del aspecto estético

El problema no es Gnome, ni Metacity, ni el GDM ni nada por el estilo. Encuentro escritorios por la red que son verdaderamente "bonitos" y no lo digo yo, sino una masa de usuarios importante. ¿Cuál es el problema? Que el escritorio que trae por defecto Ubuntu, aunque sea muy personalizable, tiene todavía muchos detalles que cambiar. Muchos usuarios quieren nada más arrancar la primera vez, un escritorio atractivo a la vista y con un aire moderno. Los tonos naranjas y marrones predominantes desde un comienzo, están siendo paulatinamente sustituidos por tonos más oscuros. Ese es el camino, pero todavía queda mucho para llegar tener un escritorio llamativo y sobre todo *cool* y práctico a la vez.

Aparte, algunos detalles menores como la sustitución de la barra inferior de Gnome por un *dock* al estilo de Avant Window Navigator o la instalación por defecto de un lanzado de aplicaciones, también ayudarían en esta labor.

Mayor instalación de software por defecto

Nada más instalar Ubuntu tenemos un sistema de escritorio totalmente funcional... al que le faltan ciertas cosas como los *restricted-extras* que muchos usuarios posteriormente instalan pero que, salvando problemas legales de por medio, deberían ser más accesible para el usuario novel de forma que tuviera una mejor impresión de Ubuntu. Hay un metapaquete llamado "caras B" o "B-sides" del que hablo en las noticias breves que está relacionado con esto.

Mejora del Ubuntu Software Center

Para mí es algo que no tiene mucho sentido utilizar, prefiero usar Synaptic o directamente apt, pero creo que es una muy buena iniciativa para los usuarios noveles. Acostumbrados a tener que buscar y descargar los ejecutables en sistemas como Windows, es mucho más cómodo tener un índice de software organizado e indexado como el *software center* pero que todavía tienen que mejorar para hacerlo más intuitivo y transparente para los usuarios más "verdes".

Compatibilidad y montaje por defecto de las particiones NTFS

Ubuntu es compatible con las particiones NTFS, pero si no instalamos y configuramos algún que otro programa, tendremos que montar manualmente las particiones cada vez... y al usuario que se ve obligado a tener una instalación dual de Ubuntu + Windows... este tema le molesta. Deberían automatizar el montaje desde la misma instalación.

Compatibilidad hardware

Es un tema en el que se mejora día a día, pero todavía quedan problemas graves que resolver. La culpa no es sólo de Ubuntu, Canonical o el software libre... tienen mucho que ver los fabricantes de hardware, pero siempre se podrá seguir avanzando.

Muchos otros pequeños detalles que seguro que vosotros también podríais aportar y que harían que esta distribución en concreto, y Linux en general, fueran más accesibles, amables y útiles para todos. Es cierto es que esta lista cada vez tiene menos detalles... así que en breve podremos ver la versión 10.04 con sus mejoras y podremos realizarnos las mismas preguntas y respuestas.

Mandriva 2010.0

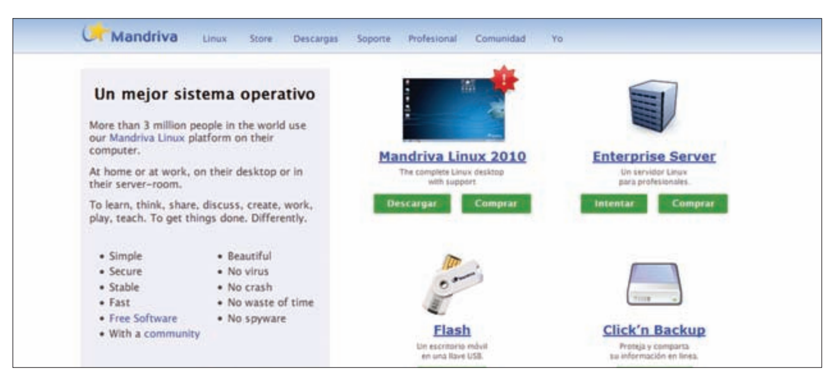
A estas alturas ya sabréis que está en la calle la nueva versión de nuestra distribución favorita, esto es Mandriva 2010.0. Esta nueva versión de Mandriva ha sorprendido a muchos, tanto por sus cambios a nivel externo como lo que nos encontramos cuando levantamos el "capó" de la distribución y vemos el funcionamiento interno de ésta.

Pero vamos a desgramar un poco lo que nos encontramos en Mandriva 2010.0. En principio nos encontramos con cuatro ediciones disponibles de esta distribución: Mandriva One (LiveCd instalable con KDE, GNOME o Xfce), Mandriva Free (1 DVD), Mandriva Powerpack (2 DVD) y Mandriva Flash (1 USB autoarrancable de 8 GB).

Mandriva 2010.0 ofrece KDE 4.3.2 y GNOME 2.28, además de Mozilla Firefox 3.5 y OpenOffice.org 3.1.1 entre otros. Además utiliza la versión 2.6.31 del kernel de Linux parcheado por Mandriva. En esta versión se ha realizado un esfuerzo especial para tener un completo soporte de hardware para un amplio rango de netbooks, cubriendo todo el rango de productos ofrecidos por Asus Eee PC, el Acer Aspire One y muchos otros. Pero lo más destacado en muchos de los foros y blogs de internet ha sido lo siguiente:

Nuevo diseño del instalador. El interfaz del instalador se ha mejorado mucho, sobre todo la herramienta de particionamiento de disco duro, siendo ésta mucho más amigable para el usuario y menos confusa para el usuario novel. Escritorio Moblin. Ya hemos hablado de este escritorio anteriormente. Como recordareis este escritorio ha sido diseñado para plataformas móviles, principalmente netbooks pero también puede ser utilizado cuando se quiera un ambiente simple y ergonómico para el uso diario en su ordenador de sobremesa. Para tenerlo disponible solo tiene que instalar el paquete task-moblin. Mandriva ha sido la primera distribución en incluir el escritorio Moblin en sus repositorios. Cuentas de invitado. Se pueden crear cuentas de invitado si no queremos que un conocido o un usuario ocasional de nuestro ordenador utilice nuestra cuenta de usuario.

Nuevo compilador gcc. Mandriva, como otras muchas distribuciones, ha realizado el cambio a la versión 4.4 del compilador gcc, beneficiándose de las mejoras en cuanto el código generado, errores, velocidad de compilación y otras mejoras que acompañan a esta versión de este compilador. Mejora en el tiempo de arranque. Mandriva ha mejorado mucho el tiempo de arranque, en muy pocos segundos tenemos a nuestra Mandriva preparada para trabajar. También se ha reducido considerablemente los tiempos de apagado. Soporte para tarjetas 3G y Wifi mejorado. Se ha mejorado la gestión de los PIN/PUNK en las tarjetas 3G, así como la posibilidad de establecer cuotas de subida y bajada de datos en las mismas. Así mismo se ha ampliado el soporte a algunos chipset de tarjetas wifi que habían generado problemas y conflictos en anteriores versiones. Estas son solamente unas breves pinceladas sobre la nueva versión de Mandriva. En mi opinión esta versión es una gran distribución, una de las mejores que ha entregado la empresa francesa a la comunidad GNU/Linux, tanto por su cuidado aspecto y diseño, como por su potencia y estabilidad. Con esta versión creo que también se acaba con la leyenda negra de problemas e inestabilidades que acompañaban a las versiones de Mandriva terminadas en 0. Por ello os recomiendo que os bajéis esta versión de esta dirección (*http://www2.mandriva.com/es/downloads/*), la probéis y, podría apostar, la instalareis en vuestro disco duro para que sea vuestro entorno de trabajo habitual.



Mandriva Website

Nueva imagen de la Wiki de Mandriva

La wiki de Mandriva ha cambiado de imagen, siendo la nueva más moderna e intuitiva, os recomiendo que la visitéis pues encontrareis una gran cantidad de información acerca de nuestra distribución favorita.



Versiones alternativas de Mandriva

Dentro de blogdrake podéis encontrar una lista de las versiones de Mandriva no oficiales. Estas distribuciones son desarrolladas y mantenidas por las comunidades de usuarios de Mandriva. Estas versiones contemplan otros escritorios alternativos a KDE o GNOME, están orientadas a entornos particulares como el educacional, etc. Podéis encontrar la lista completa en *http://blogdrake.net/blog/drakor/versiones-de-mandriva-linux-no-oficiales-desarrolladas-y-mantenidas-por-sus-comunidades.*



Mandriva 2010.1 arranca

Los chicos de Mandriva no descansan, no acaban de sacar la nueva versión de su distribución cuando ya están preparando la nueva. Podéis contribuir en la creación de la nueva distribución de muchas formas y desde aquí os animamos a hacerlo. Podéis seguir su desarrollo y contribuir con él en: *http://wiki.mandriva.com/es/Desarrollo_de_Mandriva_2010.1.*



Goddard o Fedora 13
Ya tiene nombre la nueva versión de Fedora. Según aparece en la web de noticias de Fedora y de la mano de Carlos Sepúlveda nos ha llegado la buena nueva del nombre del próximo lanzamiento de nuestra distro favorita, y va a ser GODDARD. Elegido como es habitual en los lanzamientos de la casa, mediante votación, el apelativo Goddard recibió un total de 1177 votos, contra los 1009 de su más inmediato perseguidor Langstrom seguidos por el resto de candidatos que pasaron a la final 997 (Gloriana), 992 (Botany), 707 (Loana), 654 (Truro) y 504 (Manfredi). El elegido fue seleccionado para concursar en la elección de nombre en un primer momento en honor a Robert Hutchings Goddard, por tratarse de uno de los científicos que en la década de los años veinte, se convirtió en pionero en el campo los cohetes espaciales, que luego sería el precursor e inspiración de aquellos hombres de ciencia que llevarían al hombre a la luna.

Red Hat supera a Microsoft
No deja de ser un notición, aunque en este caso se refiere a que las acciones de la empresa del sombrero rojo de GNU/Linux han superado con creces a la que es en la actualidad la que domina en el panorama de los pc aunque en continua baja. Según fuentes especializadas las acciones de Red Hat no han dejado de subir desde 2001, concretamente se han incrementado en un 600 % aproximadamente, mientras que las de la empresa de las ventanitas de colores han caído en el mismo periodo.

Gen dominante de Red Hat
Paul Cormier, vicepresidente ejecutivo de Red Hat, la empresa responsable de la creación y desarrollo de la distribución GNU/Linux del mismo nombre ha afirmado que la citada empresa es acreedora del título que dice que tiene el 75 % del mercado comercial del sistema operativo del pingüino. Con esto se pone de manifiesto que su competidor más directo, la empresa Novell, está muy alejada en este ranking y bastante más lejos Canonical, la compañía del multimillonario sudafricano. Esto no es fruto de la casualidad ya que se trata de la empresa que más trabaja en el desarrllo del Kernel.
Por otra parte, el Director Ejecutivo de Red Hat Jim Whitehurst afirma que es necesario desarrollar aún más los integradores de sistemas como objetivo comercial, lo que no es óbice para realizar acuerdos con las grandes compañías del sector.
El resultado de todo este trabajo ha sido que la compañía del sombrero rojo ha visto incrementados sus beneficios en un 12 % sobre los resultados del año pasado.

Revisor

Una versión de Fedora a nuestro gusto ¿Por qué no? Creada por Fedora Unity y formando parte de Fedora aproximadamente desde la versión 7, es una herramienta que no es conocida por muchos de los usuarios habituales. Como digo es posible utilizarla con cualquiera de ellas desde que fue lanzada, pero lo recomendable es usar la versión más actual de la distribución. Pero, ¿Para qué sirve Revisor?, Pues bien, Revisor es una aplicación que con una interfaz gráfica, nos permite crear una distribución completamente customizada a nuestro gusto, adaptarla totalmente a nuestras necesidades, casi como si creáramos nosotros mismos nuestra propia distribución, árdua tarea si partiésemos desde cero, pero ahí está la herramienta, Revisor. Después de instalar el Sistema Operativo, nos toca instalar la aplicación que nos va a facilitar todo el proceso. Esto se puede hacer de dos formas, o bien utilizamos el modo gráfico con el menú Administración, Agregar quitar software, y aquí buscamos el paquete de "Revisor", hacemos click en aplicar y aceptamos los paquetes dependencia que han de ser instalados y ya debe estar listo.

Cabe la posibilidad de que nos de un error de SELinux, para ello lo único que debemos hacer es hacer más permisiva la directiva y listo, desaparecerá el error. Revisor nos permite crear el medio tanto para una distribución estandar como para un distribución LIVE, y dentro de éstas nos permitirá crear LIVE CD/DVD o una memoria USB, permitiéndonos en todo caso la personalización del escritorio y demás componentes de la distro. También da la opción de personalizar el repositorio, según sea una plataforma genérica i386, 64-bits o PowerPC, pero teniendo en cuenta que debe estar ejecutándose en el hardware que estamos utilizando. Ahora toca decidir qué paquetes van a formar parte de nuestra distro, pues cuantos más pongamos más espacio nos ocupará, muy importante si queremos meterla en un CD. Después pasamos a la configuración en si, definimos el idioma, el teclado y la contraseña del superusuario para el modo Live entre otros parámetros.

Para obtener más información sobre Revisor podemos dirigirnos a las siguientes webs: Revisor: *http://revisor.fedoraunity.org/* o Fedora Unity : *http://fedoraunity.com, ambas webs están en inglés.*

Más novedades en Constantine

Entre las muchas novedades que cada lanzamiento de Fedora nos acostumbra a aportar, Constantine nos deleita con las siguientes:

- En primer lugar nos haremos eco de las mejoras en el rendimiento, que ha sido optimizado para los paquetes de 32 bits en plataformas i686 y que como ya dijimos anteriormente ha dejado de tener soporte para los procesadores más antiguos, tomándose un especial interés en los nuevos Atom.
- Yum-presto ahora lo tenemos por defecto, lo que hace que las actualizaciones sean mucho más rápidas puesto que su tamaño se ve significativamente reducido. También ayuda a ello el hecho de que el formato de compresión ha pasado de ser gzip a XZ, aligerando así también el trabajo de la CPU.
- Respecto de NetworkManager las principales mejoras se han realizado en el apartado de las conexiones de banda ancha. También se ha introducido soporte para Bluetooth PAN.
- Fedora 12 incluye Theora 1.1 donde se ha mejorado notablemente tanto en formato descargable como en streaming. La colaboración entre los responsables Xiph.org y Mozilla.org hacen que Firefox sea más versátil en la actualidad a la hora de reproducir multimedia.
- Soporte para tarjetas AMD Radeon HD2400 o superiores y por supuesto también para tarjetas NVIDIA, también para el trabajo para múltiples monitores. El Displayport de Intel también es soportado. Mejoras en la virtualización de la mano de KVM. Por defecto se habilita Abrt que es una herramienta que reportará de forma automática las caidas del servidor de seguridad SELinux. Dracut, como herramienta initrd.
- Mejoras en PackageKit de la mano de los plugins que ahora si un usuario intenta ejecutar un comando de un paquete que no está instalado, permitirán la instalación de éste y otro plugin nos permitirá la instalación de plugins desde el navegador.

Interfaces gráficas con SDL

David Puente Castro (blackngel)

Si estás aburrido de escribir programas en modo consola; si has experimentado con "ncurses" y su potencia o gracia no te acaba de persuadir, pero todavía no quieres introducirte en el mundo de Glade, GTK, Borland C++ Builder o cosas por el estilo, entonces y sólo entonces puede que SDL sea lo que estás buscando.



linux@software.com.pl

SDL es una librería pensada inicialmente para la creación de videojuegos en 2D (puede ayudar en 3D junto con OpenGL). Pero este artículo no se centra en tal habilidad. Nosotros aprovecharemos esta librería para crear interfaces gráficas de usuario, más conocidas como GUI. Esta librería nos dará una sensación de Programación Orientada a Eventos.

SDL [1], o Simple DirectMedia Layer es compatible con la mayoría de Sistemas Operativos, incluyendo Linux, Windows, MacOS, las variantes BSD, y muchos otros. Desarrollaremos el código bajo Linux y mostraremos cómo compilar los programas correctamente; no obstante, portar todo lo aquí descrito a cualquier otro sistema acaba resultando en algo trivial.

Una de las características más importantes de SDL es su división en subsistemas tales como vídeo, audio, eventos, cdrom, red, hilos (threads), manejo de texto y más... varios de estos subsistemas forman parte de extensiones que han venido al rescate de la, a veces arcaica, base de SDL. Se explicarán en su momento y se indicarán las opciones de compilación correspondientes. Gracias a esta forma de trabajar, nosotros podemos elegir los que nos interesen y empezar a desarrollar nuestras aplicaciones inmediatamente.

Para terminar esta introducción, quisiera mencionar que esta serie de artículos que seguirán no son ni mucho menos una guía del tipo "paso a paso" (aunque al principio lo parezca), sino que los listados de código que se irán mostrando están sacados de una aplicación real que yo mismo he programado para uso personal. Esto resulta muy efectivo, pues todas las piezas de código tienen un claro objetivo y adquieren una cierta estructura que tú mismo puedes utilizar en tus propios programas.

No obstante, si quieres ir accediendo a algunas ideas básicas sobre SDL, puedes acceder también a la referencia citada en [2].

Iniciando SDL

Bien, empezaremos por añadir a nuestro programa la cabecera principal:

```
#include "SDL/SDL.h"
```

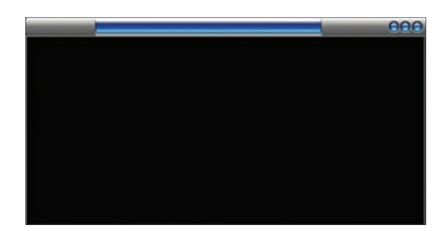


Figura 1. Imagen ventana

Definiremos una superficie principal que representará la pantalla principal de la aplicación durante la ejecución del programa:

```
SDL_Surface *pantalla;
```

Ahora, lo principal es iniciar el sistema SDL en sí. La siguiente función nos proporciona esta facilidad:

```
atexit(SDL_Quit);
if (SDL_Init(SDL_INIT_AUDIO|SDL_
INIT_VIDEO) < 0) {
        fprintf(stderr, "Error al
iniciar SDL: %s\n", SDL_GetError());
        exit(-1);
}
```

La primera llamada puede haber llamado tu atención, pero es muy sencillo. La función "atexit( )" registra el nombre de las funciones que serán llamadas antes de la finalización del programa. Como puedes observar, al iniciar el sistema básico también podemos arrancar otros subsistemas mediante el uso de constantes combinadas con un OR lógico. Podemos utilizar otros como:

```
SDL_INIT_CDROM
SDL_INIT_TIMER
SDL_INIT_JOYSTICK
```

O iniciarlos todos con:

```
SDL_INIT_EVERYTHING
```

Si has olvidado iniciar un subsistema y no lo has indicado en esta función, todavía puedes arrancarlo de esta forma:

```
if (SDL_InitSubSystem(SDL_INIT_
JOYSTICK) == -1) {
        fprintf(stderr, "No se
puede iniciar el joystick %s\n",
SDL_GetError());
        exit(1);
}
```

Podrás detenerlos análogamente con la función "SDL_QuitSubSystem( )". Establezcamos seguidamente el modo de vídeo, que creará nuestra ventana principal con la resolución y profundidad de color que indiquemos y ciertos atributos que explicaremos a continuación:

```
pantalla = SDL_SetVideoMode(1024,
768, 24, SDL_ANYFORMAT | SDL_
DOUBLEBUF);
```

Esta función devolverá "null" en caso de error. No olvides comprobarlo si no quieres sufrir errores desastrosos. En dicho caso puedes llamar a SDL_GetError( ) y SDL_Quit( ) para limpiar todo antes de llamar a exit( ) para terminar el programa definitivamente. Presento en la Tabla número 1 una lista con las constantes que se le pueden pasar a la función anterior como cuarto parámetro seguidas de sus correspondientes significados (Tabla 1).

Las constantes anteriores puedes encontrarlas directamente en la cabecera correspondiente; no obstante, yo he cogido las descripciones utilizadas en este documento [3]. Existen otras 6 constantes, pero son de uso interno y no las podrás establecer con la función anterior. Puedes estudiar más en <SDL/SDL_video.h>. Por último, si quieres definir el título de la ventana de tu aplicación, puedes hacerlo mediante:

```
SDL_WM_SetCaption("Hack The World",
NULL);
```

El segundo parámetro es el nombre de un icono opcional, puedes obtener estos valores con la análoga "SDL_WM_GetCaption( )" y establecer por separado un icono con "SDL_WM_SetIcon(SDL_LoadBMP("/home/usuario/icono.bmp"), NULL)". Incluso puedes minimizar la pantalla durante la ejecución del programa mediante la función "SDL_WM_IconifyWindow( )".

Si quieres ver cómo funciona tu primera ventana, puedes utilizar un código general de compilación como este:

```
$ gcc prog.c -lSDL -o prog
o
$ gcc proc.c `sdl-config --cflags`
`sdl-config --libs` -o prog
```

Vale, hasta aquí tenemos lo básico para iniciar el sistema y comenzar a cargar gráficos (imágenes).

Eventos

Esta es, sin duda alguna, la parte más importante de una aplicación creada con SDL. Se inicia siempre junto al sistema de vídeo y es en resumen la parte del sistema que nos permite interactuar con la aplicación, ya sea mediante el teclado, el ratón, un joystick u otros relacionados más directamente con el sistema operativo.

Como aperitivo mostraré en el listado 1 el bucle principal (main loop) que yo suelo utilizar para controlar los eventos en mis aplicaciones. Luego daré una pequeña aclaración sobre su estructura y funciones, dando paso, ya por último, a las siguientes secciones que explicarán particularmente el caso de cada dispositivo de entrada.

Empecemos por el principio. Lo más básico es el bucle while que estará siempre ite-



Figura 2. Esquema de ventanas en una aplicación particular

rando hasta que la variable "terminar" tenga un valor positivo. Seguidamente viene la función más importante. No me queda más remedio que explicar en este momento las funciones de captura de eventos que SDL nos facilita. Son tres:

- *SDL_WaitEvent( )* → Esta función espera por la llegada de un evento, ya sea un clic, una pulsación de teclado, la petición de cierre de la ventana u otro cualquiera. Se puede decir que esta función es "bloqueante" pues nada ocurrirá mientras un evento no se produzca y demos una respuesta al mismo.
- *SDL_PollEvent( )* → Muy parecida a la anterior, pero esta es más usual en los videojuegos. En estos entornos, se precisa que la aplicación siga realizando tareas en segundo plano aun a pesar de que no existan eventos a los que responder. Esta llamada no es bloqueante y ese es el motivo.
- *SDL_PumpEvents( )* → Esta función nos permitirá acceder directamente al estado de un dispositivo para conocer si algún evento se ha producido en ese mismo instante. Su uso es menos habitual y puede que sólo la veas ocasionalmente.

Lo que vemos a continuación es una sentencia "switch" que maneja el valor almacenado en el campo "type" de la estructura "SDL_Event". De este modo sabremos exactamente qué evento se ha producido. Puedes entonces introducir tantas sentencias "case" como constantes hay definidas en la enumeración "SDL_EventType" declarada en el archivo de cabecera <SDL/SDL_events.h> En este caso hacemos lo siguiente:

- Para "SDL_QUIT", que resulta de hacer clic en la [X] de la ventana, activamos la variable 'terminar' que provoca la salida inmediata del bucle "while".
- Para "SDL_MOUSEBUTTONDOWN", que resulta de hacer clic en cualquier región de la ventana, reproducimos un sonido (se verá en una sección posterior), y ejecutamos una acción según dónde se haya hecho clic (se verá en la sección 3.1.1).
- Para "SDL_MOUSEMOTION", que resulta de haber movido el ratón dentro de la ventana, nos comportamos prácticamente como en el evento anterior pero sin reproducir sonido alguno.
- Para "SDL_KEYDOWN", que resulta de presionar cualquier tecla, reproducimos

Listado 1. Bucle principal de eventos

```
SDL_Event evento;
int terminar = 0;

while (!terminar) {
        SDL_WaitEvent(&evento);
        switch (evento.type) {
                case SDL_QUIT:
                        terminar = 1;
                        break;
                case SDL_MOUSEBUTTONDOWN:
                        sonidos(1);
                        /* printf("\nX=%d - Y=%d\n",
evento.button.x, evento.button.y); */
        terminar = accion(evento.button.x, evento.button.y);
                break;
        case SDL_MOUSEMOTION:
        motion_cursor(evento.motion.x, evento.motion.y);
                break;
        case SDL_KEYDOWN:
                sonidos(2);
                mkeys(evento);
                break;
        default:
                break;
        }
}
```

Listado 2. Gestión de zonas de clicado

```
int accion(Uint16 X, Uint16 Y)
{
        /* BOTON 1 */
        if ((X >= 915 && Y >= 685) && (X <= 1024 && Y <= 718))
{
        funcion01(arg1, arg2, ...);
        }
        /* BOTON 2 */
else if ((X >= 915 && Y >= 648) && (X <= 1024 && Y <= 682)) {
                funcion02(arg1, arg2, ...);
        }
        /* BOTON 3 */
else if ((X >= 915 && Y >= 614) && (X <= 1024 && Y <= 646)) {
                funcion03(arg1, arg2, ...);
        }
        /* BOTON SALIR */
else if ((X >= 915 && Y >= 578) && (X <= 1024 && Y <= 610)) {
        return 1;
        }
return 0;
}
```

un sonido y ejecutamos una función que se encarga de controlar qué pulsación hemos realizado exactamente y realizar un cometido según corresponda.

Habrás observado una sentencia "printf( )" convenientemente comentada. Pues bien, esta instrucción te será prácticamente imprescindible en la etapa de desarrollo de tus programas. Imprime las coordenadas donde has hecho clic, y esto será más que necesario cuando quieras conocer la posición de un objeto situado en pantalla y definir "regiones de clicado" de un modo veloz.

Listado 3. Gestión del teclado

```
void mkeys(SDL_Event ev)
{
        int shift = 0;

        int altgr = 0;

        if (ev.key.keysym.mod & (KMOD_LSHIFT|KMOD_RSHIFT))
                shift = 1;
        else if (ev.key.keysym.mod & (KMOD_LALT|KMOD_RALT))
                altgr = 1;

        switch (ev.key.keysym.sym){
                case SDLK_ESCAPE:
                        terminar = 1;
                        break;
                case SDLK_BACKSPACE:
                        borrar();
                        break;
                ...
                ...
                case SDLK_a:
                        if (shift)
                                escribir('A', 1);
                        else
                                escribir('a', 1);
                        break;
                case SDLK_b:
                        if (shift)
                                escribir('B', 1);
                        else
                                escribir('b', 1);
                        break;
                ...
                ...
                case SDLK_1:
                        if (shift)
                                escribir('!', 1);
                        else if (altgr)
                                escribir('|', 1);
                        else
                                escribir('1', 1);
                        break;
                ...
                ...
                default:
                        break;
        }
}
```

Pasemos ahora a describir las acciones de comportamiento que utilizaremos para cada dispositivo.

El ratón

Como acabas de ver hace tan solo unos instantes, cuando un evento "click" se produce, las coordenadas de la pulsación son almacenadas en el elemento "button" de la unión "SDL_Event". He dicho que "button" es un elemento y no una variable porque en realidad es otra estructura. Lo que es más; salvo uno, todos los elementos de "SDL_Event" son estructuras que controlan todas las propiedades de cada evento.

Pero no nos vayamos del tema. Si sigues echando un vistazo a la estructura "SDL_MouseButtonEvent", podrás ver otros elementos, como cuál de los dos botones del ratón se ha pulsado, o si el botón pulsado está bajando o subiendo, aparte de otras más.

Una función típica que ejecute acciones según las coordenadas donde hayas pulsado suele tener la siguiente estructura que mostramos en el listado número 2.

La función puede llegar a complicarse tanto como desees, pero al fin y al cabo siempre acaba teniendo una estructura similar. Puedes pensar ahora para qué necesitas una función que controle no sólo los clics, sino también el movimiento del ratón sobre la ventana. Pues es bastante fácil; imagínate que tienes unas imágenes representando botones (se verá más adelante), podrías desear que los botones se iluminen cada vez que pasas por encima de ellos.

Se me ocurre ahora por ejemplo otra forma de cómo podría gestionarse en C++. Se crea una clase botón, que tiene como propiedades tanto la imagen que representa el botón cuando está pulsado como cuando no lo está, así como la posición que ocupa dicho botón en la pantalla. Se tiene entonces una matriz conteniendo tantos elementos como objetos se hayan creado (uno por botón), y la función "accion( )" compara uno a uno los objetos de dicha matriz para ver con cuál coincide con las coordenadas de la pulsación con respecto a la posición del botón.

Con todo esto quiero decir que existen mil maneras distintas para plantear y solucionar el problema, tú debes encontrar la que mejor se adapte a tu situación; pero ojo, debes prestar especial atención a la hora de crear esta función. Puede parecer invisible, pero esta función se ejecutará cada vez que el cursor del ratón se mueva un solo píxel en cualquier dirección. Esto quiere decir que la función puede ejecutarse cientos de veces

por segundo. Si los condicionales no están bien definidos y tiene que recorrer más de los que debería, el rendimiento podría verse seriamente afectado.

No tendrás problema alguno en desarrollar la tuya propia, pero quizás veamos algo más adelante cuando tratemos con la representación de botones o menús.

El teclado

Para controlar las pulsaciones del teclado accederemos a un miembro de "SDL_Event" llamado "key" que es una estructura "button" y que a su vez contiene otra estructura más con el nombre "SDL_keysym" cuyo elemento más importante es el miembro "sym" que, a pesar de que penséis que os estoy tomando el pelo, es una última estructura que define la totalidad de las constantes referentes a las posibles teclas pulsadas. Según la tecla pulsada tú podrás hacer lo que creas conveniente. Yo mostraré en el listado número 3 un ejemplo que llama a una función "escribir( )" para cada tecla pulsada cuyo objetivo es escribir el carácter correspondiente en la representación de una consola (o shell) que estudiarás en una secuela posterior de este artículo.

Tan solo es un extracto de lo que sería el código real completo. Como ya he dicho, lo importante es comprender el procedimiento que seguimos para llevar a cabo nuestros objetivos. Posteriormente echaremos un vistazo a las funciones que en este caso he utilizado. Es muy bueno observar el uso que hemos hecho del elemento "mod" de la estructura "SDL_keysym". Siempre comprobamos si cada vez que pulsamos una tecla lo hacemos simultáneamente con SHIFT o ALT(GR).

Imágenes

Lo principal es que conozcamos qué es una "superficie". En realidad es una estructura que controla todos los píxeles de una imagen o región de la pantalla así como su profundidad de color y otro tipo de propiedades. Nos sobrará con saber que se definen siempre igual que la superficie principal:

```
SDL_Surface *imagen;
```

Otra estructura básica que debemos estudiar es "SDL_Rect", sirve para definir una región en la pantalla indicando sus coordenadas iniciales y el ancho/alto de la misma. Se acostumbra a utilizar así:

```
SDL_Rect rect = (SDL_Rect) {300,
300, 100, 100};
```

Esto define una región que comienza en las posiciones x=300, y=300, teniendo una medida de 100 píxeles tanto para el ancho como para el alto. Si los dos últimos valores son iguales a 0, cuando dibujemos la superficie se igualarán automáticamente al ancho y alto de la misma. Podemos también acceder a sus elementos individualmente de esta forma:

```
rect.x = 300;
rect.y = 300;
rect.w = 100;
rect.h = 100;
```

SDL nos proporciona una función básica para la carga de imágenes, que es conocida como:

```
imagen = SDL_LoadBMP("/home/usuario/
file.bmp");
```

Devuelve siempre el valor "null" cuando no ha podido cargar el archivo. Pero esta función es pobre y no soporta otros formatos. Es aquí donde entran en juego las librerías auxiliares. En este caso SDL_image. Antes de pasar a estudiar el uso de esta librería debemos conocer algunas funciones más para el manejo básico de las superficies. Por ejemplo, ahora que ya tenemos cargada una superficie en la variable "imagen", la pregunta es: ¿cómo dibujarla en pantalla? "SDL_BlitSurface( )" al rescate. Mostraremos cómo se usa y luego explicaremos sus argumentos:

Listado 4. Representación de ventanas

```
SDL_Surface *v1;
void carga_v1(int put)
{
        v1 = IMG_Load(IMAGES"v1.png");
        if (!v1) {
        fprintf(stderr, "No se pudo cargar v1.png\n");
                SDL_Quit();
                exit(-1);
        }
        if (put)
                put_v1();
}
void put_v1(void)
{
        SDL_Rect r1;
        r1 = (SDL_Rect) {0, 0, 705, 375};
        SDL_FillRect(pantalla, &r1, SDL_MapRGB
           (pantalla->format,0,0,0));
        r1 = (SDL_Rect) {0, 0, 0, 0};
        SDL_BlitSurface(v1, NULL, pantalla, &r1);
        SDL_Flip(pantalla);
}
```

Listado 5. Desaparición diagonal de ventana

```
void quitar_v1(void)
{
        SDL_Rect pos;
        int x=0, y=0;
        while(x < 400 && y < 400){
                pos = (SDL_Rect) {0, 0, 705, 375};
                SDL_FillRect(pantalla, &pos,
                SDL_MapRGB(pantalla->format,0,0,0));
                pos = (SDL_Rect) {0-x, 0-y, v1->w, v1->h};
                SDL_BlitSurface(v1, NULL, pantalla, &pos);
                SDL_Flip(pantalla);
                x += 20;
                y += 20;
                SDL_Delay(20);
        }
        is_v1 = 0;
}
```

```
SDL_BlitSurface(imagen, NULL,
pantalla, &rect);
```

- La superficie que queremos dibujar.
- La porción de la superficie que queremos dibujar (SDL_Rect). Un valor null dibuja la superficie completa.
- La superficie sobre la que dibujaremos.
- Las coordenadas donde imprimiremos la superficie a pintar (SDL_Rect).

No siempre desearemos dibujar nuestras superficies o imágenes directamente sobre la pantalla principal; pero quedas advertido, si realizas un "blit" sobre cualquier otra superficie, luego estarás obligado a realizar el "blit" de esta última sobre la pantalla principal. Muy bien, si has llegado hasta aquí y has intentado dibujar una imagen propia sobre la ventana principal, te preguntarás por qué ésta no es visible y parece que la ejecución de tu aplicación ha fallado. No te precipites, aquí tienes la solución:

```
SDL_Flip(pantalla);
```

Esta función provoca que la representación gráfica que has creado en memoria hasta el momento, sea volcada en pantalla. Podrías decir vulgarmente que estas "actualizando la pantalla" si te sientes más cómodo.

Tabla 1. Constantes de inicialización

Constante	Significado
SDL_SWSURFACE	Superficie en memoria del sistema.
SDL_HWSURFACE	Superficie en memoria de vídeo.
SDL_ASYNCBLIT	Actualización asíncrona.
SDL_ANYFORMAT	Fuerza el uso de los bpp de la surface actual. Hay que usarlo cuando queramos crear la superficie en una ventana.
SDL_HWPALETTE	Da a SDL acceso exclusivo a la paleta de color.
SDL_DOUBLEBUF	Solo válido con SDL_HWSURFACE. Técnica del "doble buffer".
SDL_FULLSCREEN	Visualización a pantalla completa.
SDL_OPENGL	Crea un contexto OpenGL.
SDL_OPENGLBLIT	Igual a la anterior, pero SDL hace el renderizado 2D.
SDL_RESIZABLE	La ventana puede cambiar de tamaño.
SDL_NOFRAME	Crea una ventana sin borde.

Librería SDL_image

La función más importante que proporciona esta librería es:

```
SDL_Surface *
IMG_Load(const char *file)
```

Puedes usarla tal y como lo has hecho con "SDL_LoadBMP( )". Pero entonces, ¿cuál es la diferencia? Una y muy grande. Soporta los siguiente formatos:

```
BMP, PNM, XPM, LBM, PCX, GIF, JPG,
PNG y TGA.
```

Puedes incluir esta librería muy fácilmente añadiendo "-lSDL_image" a tus opciones de compilación normales. No hay más que decir, creo que está más que claro que ésta es la función que empezarás a utilizar a partir de ahora.

Ventanas

La cruda realidad es que SDL no comprende entidades como ventanas o botones. No tiene estructuras para manejar este tipo de objetos y eso supone un paso hacia atrás para nosotros. En SDL todo es apariencia, y como tal, nuestro objetivo es utilizar una imagen para darle la

apariencia de una ventana y cierto comportamiento que se asemeje lo suficiente.

Lo normal es utilizar siempre una imagen en formato "PNG", dado que soporta las transparencias y ese es un aspecto muy útil a la hora de crear una interfaz atractiva y eficiente. Las imágenes en este formato tienen una alta compresión y ocupan relativamente poco espacio. Un editor como "GIMP" puede ayudarte mucho en tareas como esta. Pongo un ejemplo en la ilustración número 1.

Encontrar una imagen con el aspecto de una ventana es tan fácil como navegar durante unos minutos por la web o simplemente sacar una captura de pantalla de una ventana de tu PC y editarla posteriormente para borrar todo su contenido, dejando únicamente el marco de la misma.

Como aquí no vamos a ver la posibilidad de arrastrar una ventana por la pantalla, que podría tornarse en una tarea infernal (pues requiere el repintado continuo de todas las superficies por las que ésta pasa), definiremos unas regiones en las que interactuarán nuestras ventanas individualmente. Lo lógico sería situar las ventanas en cada una de las esquinas de la pantalla. En la ilustración número 2 muestro el aspecto de un programa que yo mismo he creado cuyo objetivo es analizar hosts objetivos (recaba alguna información sobre el sistema y marca su localización en un mapa mundial, es decir, un traceroute visual, además de poseer algún que otro método de ataque).

En esta captura se ve claramente cómo hemos creado, utilizando sólo imágenes, varios elementos que podrías precisar en una aplicación (ventanas, botones, consola, etc...).

Personalmente utilizo dos pequeñas funciones para cargar y mostrar en pantalla una ventana. Las expongo en el listado número 4 y las explico a continuación.

Te preguntarás por qué no poner las dos funciones en una. El motivo es el siguiente. La primera vez que entres en el programa llamarás a la función "carga_v1( )" con un argumento positivo; esto cargará la imagen (ventana) y la mostrará en pantalla. Durante el resto de la ejecución del programa, cuando realices un "blit" sobre esta ventana, llamarás simplemente a "put_v1( )", que borrará la región y repintará la ventana nuevamente con los cambios realizados. Sólo llamaremos a "carga_v1( )" cuando queramos obtener una ventana limpia, sin modificaciones.

Bien, imagínate ahora que ya tenemos la imagen de la ventana cargada en la esquina superior izquierda de la ventana. Imagínate también que esa imagen tiene dibujados dos cuadrados o circunferencias (o lo que sea) representando los botones de cierre, minimizado, etc...

Tal como se ha explicado en una sección anterior, podemos definir una "región de clicado" que realice una operación al hacer clic en esta zona. Podríamos añadir a la función "accion( )" algo como:

```
/* BOTON "CERRAR" DE V1 */
if ((X >= 642 && Y >= 6) && (X <=
662 && Y <= 26)) {
if (is_v1)
quitar_v1();
}
```

Lo más fácil es que cuando hagamos clic en este "botón", la ventana deje de ser visible ya sea dibujando un rectángulo negro encima, haciendo que vaya desapareciendo desplazándose hacia la izquierda, hacia arriba o incluso en diagonal. Nosotros, que somos así de atrevidos, y para demostrar que con poco código se pueden crear efectos atractivos, vamos a optar por esta última elección. El resultado se muestra en el listado número 5.

No te ciegues, la operación no es nada complicada. Simplemente vamos restando 20 a las coordenadas iniciales (0,0) y redibujamos la ventana en cada vuelta. Entremedias dibujamos siempre un rectángulo negro que cubra toda la zona. Es esta secuencia de "quitar y poner" la que nos ofrece una sensación real de movimiento. La ventana irá desapareciendo diagonalmente hasta perderse fuera de la región visible de la pantalla.

"SDL_Delay( )" hace una función similar a "usleep( )" que es detener el programa la cantidad de milisegundos especificados como único argumento. Puedes incrementar esta cifra si deseas un movimiento más lento o bajarla si deseas el efecto contrario.

Como puedes ver, utilizamos una última variable global "is_v1" que usamos como si de un dato booleano se tratase y que nos permitirá saber en qué estado se encuentra la ventana (abierta/cerrada o visible/oculta). Deberías te-

Listado 6. Carga de botones

```
void carga_botones(int op)
{
        switch (op) {
                case 0: { /* Botones normales */
                        boton0 = IMG_Load(IMAGES
                        "boton0.png");
                        boton1 = IMG_Load(IMAGES
                        "boton1.png");
                        boton2 = IMG_Load(IMAGES
                        "boton2.png");
                        boton3 = IMG_Load(IMAGES
                        "boton3.png");
                        break;
                }
                case 1: { /* Botones iluminados */
                        boton0 = IMG_Load(IMAGES
                        "boton0_ilu.png");
                        boton1 = IMG_Load(IMAGES
                        "boton1_ilu.png");
                        boton2 = IMG_Load(IMAGES
                        "boton2_ilu.png");
                        boton3 = IMG_Load(IMAGES
                        "boton3_ilu.png");
                        break;
                }
        }
        if (!boton0 || !boton1 || !boton2 || !boton3) {
                fprintf(stderr, "Don't load some
                                button image\n");
                SDL_Quit();
                exit(-1);
        }
}
```

ner una lista de botones en la parte inferior de la ventana, o en aquel sitio que tú elijas, que te permita traer de nuevo las ventanas que has cerrado (cómo crear botones simulados será el tema del siguiente apartado).

Hacer aparecer la ventana de la misma forma en que ha desaparecido te lo dejamos como deberes, al fin y al cabo es ir controlando las coordenadas de modo que la ventana aparezca de forma diagonal mostrando cada vez una porción mayor hasta que se muestre completa. La operación es muy parecida a la anterior. Recuerda también que las cifras van a depender siempre de tu caso en particular y que deberás ajustarlas en su momento.

La única diferencia en esta ocasión es que es muy improbable que la última iteración del bucle termine colocando la ventana en sus coordenadas exactas. Es por ello que se debe controlar cuando la ventana se acerca a esta posición y acabar de colocarla nosotros mismos en el lugar correspondiente. Ten esto muy en cuenta cuando te pongas a escribir el código.

Botones

Para practicar con la representación de botones situaremos un botón en la esquina inferior derecha que abrirá un menú cuando hagamos clic en él. También veremos cómo crear un efecto de iluminado cuando pasemos por encima de cada uno de los botones. Esto les dará una sensación de volumen y aumentará notablemente su credibilidad. Debo advertir que en la primera parte de este artículo sólo mostramos cómo dibujar el botón en pantalla y cómo se puede interactuar con él; las funciones que abren un nuevo menú y muestran el efecto de iluminado las dejaremos para la siguiente parte por motivos de espacio.

Los botones deben ser sin duda unas de las imágenes más abundantes en la telaraña de la World Wide Web. Incluso si te molestas en buscar un poco en Google, encontrarás unos pequeños programas que te permitirán crear, online, tus propios botones personalizados. Recuerda utilizar formato "PNG" preferentemente. Si utilizas botones con esquinas redondas este requisito se vuelve casi imprescindible.

La primera función que debemos definir, se encarga de cargar las imágenes en las superficies correspondientes. Según el argumento, se cargarán los botones normales o los que hayamos creado con efecto iluminado. Observa el listado 6 para ver cómo procedemos; piensa que podrías utilizar otro "case" para cargar botones, por ejemplo, con efecto de pulsado.

Comprobamos al final que todas las imágenes han podido ser cargadas correctamente y evitamos así caídas inesperadas. La siguiente función se encarga de poner los botones que serán estáticos, es decir, los que no forman parte del menú interno y serán visibles durante toda la ejecución. En nuestro caso sólo pondremos uno en la esquina inferior derecha, pero podríamos poner tres en horizontal y hacer que cada uno abriera un menú diferente siguiendo los pasos que aquí explicaremos (ver Listado 7).

Puedes ver cómo llamamos antes de nada a "carga_botones( )" con un valor de 0. Mientras no interactuemos con ellos, deben de estar en estado normal. Para lograr recrear la apertura de un menú, primero tenemos que definir una "región de clicado" en torno a "boton0". Como dije antes, la instrucción "printf( )" que imprime las coordenadas puede servirte de mucho. Haz clic cerca de la esquina superior izquierda de "boton0", anota la posición en un papel y repite el proceso para su esquina inferior derecha. Hecho esto, vete a la función "accion( )" que ya describimos anteriormente y añade algo como se muestra en el Listado 8. Como dije al principio de esta sección, por motivos de espacio aquí no vamos a mostrar cómo abrir o cerrar el menú (esto se verá en el siguiente número); no obstante, tú mismo puedes intentarlo con todo lo que ya sabes. Al fin y al cabo, abrir un menú gráficamente sólo se trata de dibujar otros botones encima del que has pulsado y cerrarlo significa simplemente dibujar un recuadro negro encima que los borre. Eso sí, siempre hay que tener una variable global que controle si el menú está abierto o cerrado, ¿por qué? muy simple; porque si no controlásemos ese detalle e hiciésemos clic en la zona donde supuestamente debería haber un botón, el programa ejecutaría las acciones que hubieses programado para él aunque el menú estuviera cerrado.

Audio

Realizar tareas con el subsistema de audio que proporciona directamente la librería SDL es posible, pero puede convertirse en toda una hazaña digna de comentar. Es por ello que la dejaremos a un lado centrándonos en la implementación de la siguiente librería auxiliar: SDL_mixer.

Librería SDL_mixer

Compilar un programa que utilice esta librería es tan fácil como añadir el archivo de cabecera <SDL/SDL_mixer.h> y ayudarse de

Tabla 2. Funciones de SDL_mixer

Función	Significado
Mix_PlayChannelTimed( )	Igual que la anterior pero lleva un último argumento que indica cuántos milisegundos se reproducirá.
Mix_FadeInChannel( ) y Mix_FadeInChannelTimed( )	Análogas a las 2 anteriores pero va subiendo el volumen del sonido de modo gradual.
Mix_Pause(channel), Mix_Resume(channel), Mix_HaltChannel(channel)	Más que obvios: Pausar, reanudar y detener. "channel" es un int.
Mix_Playing(int channel) y Mix_Paused(int channel)	Funciones de consulta; con ellas obtienes el estado actual de un canal.

Tabla 3. Funciones relacionadas con el CD-ROM

Función	Significado
int SDL_CDNumDrives(void)	Devuelve el número de dispositivos CD-ROM.
const char * SDL_CDName(int drive)	Devuelve el nombre de un dispositivo.
SDL_CD * SDL_CDOpen(int drive)	Abre un dispositivo de CD-ROM.
CDstatus SDL_CDStatus(SDL_CD *cdrom)	Devuelve el estado de un CD-ROM.
int SDL_CDPlayTracks(SDL_CD *cdrom, int start_track, int start_frame, int ntracks, int nframes)	Reproduce tantas pistas como se le indique en "ntracks" empezando en "star_track".
int SDL_CDPlay(SDL_CD *cdrom, int start, int length)	Reproduce un CD-ROM desde el frame "start" hasta la cantidad indicada por "lenght".
int SDL_CDPause(SDL_CD *cdrom)	Pausa la ejecución de una pista.
int SDL_CDResume(SDL_CD *cdrom)	Continúa la ejecución de una pista.
int SDL_CDStop(SDL_CD *cdrom)	Detiene la ejecución de una pista.
int SDL_CDEject(SDL_CD *cdrom)	Abre la bandeja del CD-ROM.
SDL_CDClose(SDL_CD *cdrom)	Cierra el dispositivo CD-ROM.

Listado 7. Dibujar botones

```
void poner_botones(void)
{
        SDL_Rect pos;
        int x=0, y=0;
        carga_botones(0);
        pos = (SDL_Rect) {910, 710, 0, 0};
        SDL_BlitSurface(boton0, NULL, pantalla, &pos);
        SDL_Flip(pantalla);
}
```

Listado 8. Acción para "boton0"

```
/* BOTON 0 */
else if ((X >= 920 && Y >= 720) && (X <= 1024 && Y <= 768)) {
        if (is_menu)
                cerrar_menu();
        else
                abrir_menu();
}
```

Listado 9. Reproducción de sonidos

```
void sonidos(int op)
{
        u_int canal;
        Mix_Chunk *sonido;
        switch(op){
                case 1:{
                        sonido = Mix_LoadWAV("/home/usuario/
                        sonidos/sample1.wav");
                        if(sonido == NULL){
                                printf("Error: %s\n",
                                Mix_GetError());
                                return;
                        }
                        canal = Mix_PlayChannel
                        (-1, sonido, 0);
                        break;
                }
                case 2:{
                        sonido = Mix_LoadWAV("/home/
                        usuario/sonidos/sample2.wav");
                        if(sonido == NULL){
                                printf("Error: %s\n",
                                Mix_GetError());
                                return;
                        }
                        canal = Mix_PlayChannel
                        (-1, sonido, 0);
                        break;
                }
                ...
                ...
                default:
                        break;
        }
        Mix_FreeChunk(sonido);
}
```

la opción "-lSDL_mixer" a la hora de utilizar tu versión de "gcc" favorita.

Lo primero es lo primero, que es iniciar el sistema de sonido. Suelo utilizar algo como lo que verás a continuación:

```
if (Mix_OpenAudio(44100, AUDIO_S16,
2, 4096)) {
fprintf(stderr, "No se puede iniciar
SDL_mixer %s\n", Mix_GetError());
SDL_Quit();
    exit(1);
}
atexit(Mix_CloseAudio);
```

Los argumentos son estos:

- Frecuencia (en hercios) para reproducir un "sample" (*).
  (*) Yo utilizo la calidad CD, otros posibles son:
  - *11025* → Calidad teléfono.
  - *22050* → Calidad radio.
- Formato del sample (revisa las constantes en <SDL_mixer.h>).
- Número de canales (1 = mono, 2 = estéreo)
- "Chunksize", esto habitualmente siempre es 4096.

Utiliza "Mix_CloseAudio( )" para detener el sistema. Quizás te estés preguntando qué significa eso de un "Chunk". Para que lo entiendas, podríamos decir escuetamente que es la estructura donde SDL_mixer almacena un sonido para poder trabajar con él.

Ahora expondré una función que me gusta utilizar para la reproducción de sonidos y luego la explicaré en detalle para que se entienda.

"MixLoadWAV( )" es análoga a la función de carga de imágenes, carga el fichero indicado en su único parámetro y lo almacena en el chunk que hemos creado. Comprobamos siempre que la función ha tenido éxito y luego llamamos sin más dilaciones a "Mix_PlayChannel( )" cuyos argumentos son los siguientes:

- Número de canal para reproducir el sonido (*).
  (*) Un valor de "-1" para selección automática del canal.
- Sonido (chunk) a reproducir.
- Número de veces que se repetirá el sonido (*).
  (*) Un valor de "-1" para reproducir indefinidamente.

Liberamos finalmente el chunk para mejorar el rendimiento. Aquí tienes más funciones, sacadas del archivo de cabecera de esta librería. Daré una ligera explicación de cada una en la Tabla 2.

Existen muchísimas más. Si te interesa crear un reproductor de sonidos o de música queda a tu elección seguir estudiando este tema. Y hablando de "música", decir que SDL_mixer diferencia realmente entre la estructura de "un sonido" y de "una música". Tanto que reserva un canal especial para la reproducción de esta última. Los formatos que reconoce son: WAV, MP3, MOD, S3M, IT, XM, Ogg Vorbis, VOC y MIDI. Todas las funciones para manejar una estructura del tipo "Mix_Music *" son prácticamente análogas a las que controlan "chunks". No te será difícil investigar un poco por tu cuenta.

CD-ROM

Para ejecutar las funciones que se enseñarán a continuación debes incluir en tus programas la cabecera <SDL/SDL_cdrom.h>. Existen dos estructuras para el control de un dispositivo CD-ROM.

"SDL_CD" controla el estado del dispositivo, el número de pistas, la pista actual, el frame actual y un array de estructuras del tipo "SDL_CDtrack".

También tiene un identificador único para cada uno de los dispositivos. Como no me voy a poner a mostrar ejemplos sobre el uso de estos métodos. Indicaré como siempre las funciones tal y como se pueden ver en la cabecera principal y un escueto comentario en la Tabla número 3.

Conclusión

Y hasta aquí hemos llegado con la primera parte de este artículo. Tengo la esperanza de que puedas sacar provecho de todos los extractos de código mostrados y de las ideas que hemos ido desmenuzando entre frase y frase. Tampoco nos hemos cortado a la hora de enseñar el cometido de funciones que ni tan siquiera hemos utilizado, pero que sabemos de antemano pueden serte útiles en el momento más inesperado.

En la segunda parte de este artículo (todavía no sabemos si serán 2 ó 3 partes en total) tocaremos todos los temas referentes a la manipulación de texto, gráficos primitivos, funciones complementarias, efectos, y por último nos adentraremos en la creación de la representación visual de una shell, simulando su comportamiento a la hora de escribir texto sobre ella, ejecutar acciones y obtener una salida correctamente formateada; todo ello manejando tan solo gráficos esenciales.

La aventura continúa, ¿estás dispuesto a acompañarnos?

En la red

- [1] SDL
  http://www.libsdl.org
- [2] Programación con SDL
  http://www.linux-magazine.es/issue/01/programacionSDL.pdf
- [3] Programación de videojuegos con SDL
  http://www.agserrano.com/libros/sdl/%5Bebook%5DProgramacion%20de%20videojuegos%20con%20SDL.pdf
- [4] Tutorial de libSDL para la programación de videojuegos
  http://softwarelibre.uca.es/tutorialSDL/TutorialSDL-30012008.pdf

Make: compilación inteligente

Andrés Tarallo

En los proyectos simples, de pocos archivos fuente y baja cantidad de lineas de código, la compilación se hace en forma manual. Este proceso es tedioso y propenso a errores. Algunos programadores utilizan scripts en shell de UNIX, que automatizan el proceso de compilación. Esa práctica resuelve los inconvenientes planteados más arriba, pero no es una buena solución. En el caso de haber cambiado un código fuente el script compilará todos los códigos fuente necesarios para construir el binario. Se hace necesaria una solución más inteligente.



linux@software.com.pl

Make es un utilitario para construcción automatizada de programas y librerías, partiendo de su código fuente. Un archivo llamado Makefile especifica cómo compilar el programa desde los códigos fuente, los archivos a compilar y las dependencias que éstos presentan. Cuando ejecutamos el comando make, éste lee el makefile y determina qué archivos fueron cambiados y es necesario volver a compilar. De esta forma se compila sólo lo necesario, reduciendo significativamente el tiempo de compilación. Es importante destacar que al comenzar la ejecución del comando make no tiene el orden de ejecución de la compilación, ésta se determina para cada corrida. Esta característica de make es compartida por los lenguajes de programación declarativa.

Make fue creado por Stuart Feldman de los laboratorios Bell. En el año 2003 el Dr. Feldman recibió el premio "Software System Award" por la creación de make.

Es importante destacar que existen varias implementaciones de make. El código original de los laboratorios Bell estaba protegido por patentes que limitaban su uso. Todas las implementaciones comparten la filosofía y estructura, cambiando las prestaciones de las mismas. Las implementaciones más populares son:

- BSD Make: es el que habitualmente se encuentra en los sistemas operativos de la familia BSD. FreeBSD, NetBSD y OpenBSD. Entre sus características más salientes está la capacidad de usar loops y condicionales.
- GNU Make: es la implementación usual en las distribuciones Linux, sobre la que se probaron los ejemplos de este artículo.
- nmake de Microsoft.

Un Makefile Simple

La estructura de un makefile es bastante simple, tenemos objetivos y comandos que se ejecutan para ese objetivo. Cada objetivo tendrá asociada una lista de dependencias, que se deberán ejecutar con antelación, previo a ese objetivo.

El objetivo nos define el módulo o programa a crear separado por dos puntos y tendremos la lista de depen-



```
[root@s170 ~]# make
make: *** No targets specified and no makefile found.  Stop.
[root@s    ~]#
```

Figura 1. Corrida de make sin makefile

```
# quick makefile for the FAQ.
# So that both the text and html FAQ are accessible on the web
# they are checked into SVN along with the .texi source.
# Don't forget to update version.texi.
# Run 'make' and then check in all 4 files after editing the .texi source.
#
all::  lm_sensors-FAQ.html FAQ

%.html: %.texi version.texi
        makeinfo --html --no-split --number-sections $<

FAQ: lm_sensors-FAQ.texi
        makeinfo --no-headers --number-sections -o FAQ $<
"/usr/share/doc/lm_sensors-2.10.7/doc/Makefile" 14L, 454C          1,1          All
```

Figura 2. Generando archivos info con make

dencias necesarias, cada una separada por espacios. En la línea siguiente tendremos el primer comando a ejecutar, es importante que los comandos estén separados por un tabulador del comienzo de líneas. Se debe tener especial cuidado con este punto, hay editores que remplazan los tabuladores por espacios en blanco, esto lleva a que los makefiles no funcionen. Aquí tenemos un esbozo de la estructura del archivo:

```
objetivo: dependencias
        comandos
```

Para fijar ideas crearemos un makefile simple para un programa en C. Este programa se compone de 3 fuentes: main.c, core.c y loop.c. Los dos últimos archivos contienen funciones que son utilizadas por el primero de ellos. Si compiláramos esto desde la línea de comandos compilaríamos los dos últimos compilando el primero para obtener de ahí el ejecutable. En el Listado 1 tenemos un makefile para este programa.

Comentarios

Los comentarios aportan claridad al makefile. El programador podría documentar allí cómo configurar el makefile para que este corra en distintas arquitecturas. Un uso bastante frecuente es documentar los códigos fuente y actividades necesarias para compilar el programa. Un ejemplo de esta práctica lo podemos ver en el paquete UW-IMAP que en el makefile trae comentarios documentando las distintas opciones de compilación del mismo. En el Listado 2 es posible ver una porción de este makefile.

Variables

En el ejemplo del Listado 2, además de los comentarios, se introducen las variables. Podemos definir en nuestro makefile variables para facilitar el porte a otras arquitecturas y sistemas. Un ejemplo muy común es definir la variable CC con el compilador C/C++ a utilizar. Para utilizar la variable que hemos definido basta con usar su nombre entre $(y). Aquí se puede ver una definición de CC y su posterior uso en un objetivo de compilación:

```
CC=gcc
juego:
        $(CC) main.c modulo.o -o
juego
```

Las variables son expandidas para cada línea, esto puede tener una consecuencia no deseada: expansiones recursivas ad infinitum. Para evitar este problema podemos utilizar un operador de asignación distinto. En lugar de = , utilizaremos := cuando queramos evitar la expansión de una variable. Aquí un ejemplo aplicado a la variable CC:

```
CC=gcc
CC:=$(CC) -O2
```

Es importante tener en cuenta que las variables de ambiente también están disponibles para make, si quisiéramos utilizar $HOME en un makefile bastaría con invocarla como $(HOME).

Tenemos algunas variables especiales que es útil conocer. Estas variables son asignadas dinámicamente para cada objetivo. Con ellas podríamos construir reglas genéricas,

Listado 1. Un ejemplo completo

```
programa: core
        gcc main.c core.o loop.o -o programa
core:
        gcc -c core.c loop.c
```

Listado 2. Una opción de compilación del paquete WU-IMAP

```
# IP protocol version
#
# The following IP protocol versions are defined:
# o      IPv4 support, no DNS (truly ancient systems)
# 4      (default) IPv4 support only
# 6      IPv6 and IPv4 support

IP=4
IP6=6
```

Listado 3. Ejemplo de objetivo clean

```
clean:
        @echo Removing old processed sources and binaries...
        $(SH) -c '$(RM) an ua OSTYPE SPECIALS c-client mtest imapd ipopd mailutil mlock dmail tmail || true'
        $(CD) tools;$(MAKE) clean
```

Listado 4a. Automatizacion de tareas en Shorewall con un Makefile

```
#      Shorewall Packet Filtering Firewall Export Directory Makefile - V3.3
#
#      This program is under GPL [http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt]
#
#      (c) 2006 - Tom Eastep (teastep@shorewall.net)
#
#        Shorewall documentation is available at http://www.shorewall.net
#
#        This program is free software; you can redistribute it and/or modify
#        it under the terms of Version 2 of the GNU General Public License
#        as published by the Free Software Foundation.
#
#        This program is distributed in the hope that it will be useful,
#        but WITHOUT ANY WARRANTY; without even the implied warranty of
#        MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
#        GNU General Public License for more details.
#
#        You should have received a copy of the GNU General Public License
#        along with this program; if not, write to the Free Software
#        Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

################################################################################

# Place this file in each export directory. Modify each copy to set HOST
# to the name of the remote firewall corresponding to the directory.
#
#        To make the 'firewall' script, type "make".
#
#        Once the script is compiling correctly, you can install it by
#        typing "make install".
#

################################################################################

#                             V A R I A B L E S
#
# Files in the export directory on which the firewall script does not depend
#
IGNOREFILES =  firewall% Makefile% trace% %~
#
# Remote Firewall system
#
HOST = gateway
#
# Save some typing
#
LITEDIR = /var/lib/shorewall-lite
```



sin necesidad de especificar uno por uno los ficheros a compilar. En Tabla 1 tenemos un resumen de ellas.

Reglas patrón

Al final del párrafo anterior, luego de enumerar las variables de make, se presentó la posibilidad de construir reglas genéricas a partir de las variables implícitas. Esto presenta como ventaja que el makefile va compilando los nuevos archivos del proyecto, a medida que los creamos. Como contra, podemos decir que al neófito se le dificulta la

Tabla 1. Estas variables pueden ser utilizadas para escribir reglas genéricas, que compilen una serie de fuentes sin necesidad de escribir explícitamente los nombres de los mismos

`$@`	nombre del objetivo de la presente regla
`$*`	la raíz de un nombre de archivo
`$<`	primera dependencia de la presente regla
`$^`	lista separada por espacios de cada una de las dependencias de la presente regla
`$?`	lista separada por espacios de cada una de las dependencias de la presente regla que sean más nuevas que el objetivo de la regla
`$(@D)`	la parte correspondiente al subdirectorio de la ruta del fichero correspondiente a un objetivo que se encuentre en un subdirectorio
`$(@F)`	la parte correspondiente al nombre del fichero de la ruta del fichero correspondiente a un objetivo que se encuentre en un subdirectorio

Listado 4b. Automatizacion de tareas en Shorewall con un Makefile

```
# Set this if the remote system has a non-standard modules directory
#
MODULESDIR=
#
# Default target is the firewall script
#

################################################################################

#                                T A R G E T S
#
all: firewall
#
# Only generate the capabilities file if it doesn't already exist
#
capabilities:
        ssh root@$(HOST) "MODULESDIR=$(MODULESDIR) /usr/share/shorewall-lite/shorecap > $(LITEDIR)/
capabilities"
        scp root@$(HOST):$(LITEDIR)/capabilities .
#
# Compile the firewall script. Using the 'wildcard' function causes "*" to be expanded so that
# 'filter-out' will be presented with the list of files in this directory rather than "*"
#
firewall: $(filter-out $(IGNOREFILES) capabilities , $(wildcard *) ) capabilities
        shorewall compile -e . firewall
#
# Only reload on demand.
#
install: firewall
        scp firewall firewall.conf root@$(HOST):$(LITEDIR)
        ssh root@$(HOST) "/sbin/shorewall-lite restart"
#
# Save running configuration
#
save:
        ssh root@$(HOST) "/sbin/shorewall-lite save"
#
# Remove generated files
#
clean:
        rm -f capabilities firewall firewall.conf reload
```

lectura y comprensión de éste. En el Listado 6 mostramos un ejemplo que ilustra claramente este concepto.

Esta regla se aplica a todos los objetivos con extensión .o (objetos), que tienen como dependencias todos los archivos fuente (extensión .c). Serán compilados con $CC con las directivas de compilador $CFLAGS, generando el objetivo ($@, un archivo con extensión .o). Es de destacar el uso de la variable $<, que corresponde al nombre de la primera dependencia del objetivo, con lo que mantenemos una compilación inteligente; compilando solo aquellos archivos que han cambiado.

Ejemplo de reglas patrón:

```
%.o : %.c
        $(CC) $(CFLAGS) $< -o $@
```

Reglas virtuales

Es usual encontrar en los makefiles reglas que no generan código, pero realizan alguna tarea del proyecto. Tal es el caso de las reglas clean o install, usuales en muchos proyectos. Estas reglas son responsables de borrar los archivos generados durante la compilación e instalar los mismos, respectivamente. En el Listado 3 tenemos un ejemplo de objetivo clean, sacado desde los códigos fuente del UW-IMAP.

Estas reglas no suelen tener dependencias,por lo que podría darse un curioso efecto colateral. Si entre los archivos existiera uno llamado clean make asumirá que el objetivo está logrado. Por esto es que muchas veces el objetivo clean lleva el tag .PHONY delante de él:

```
.PHONY : clean
         rm -f *.o
```

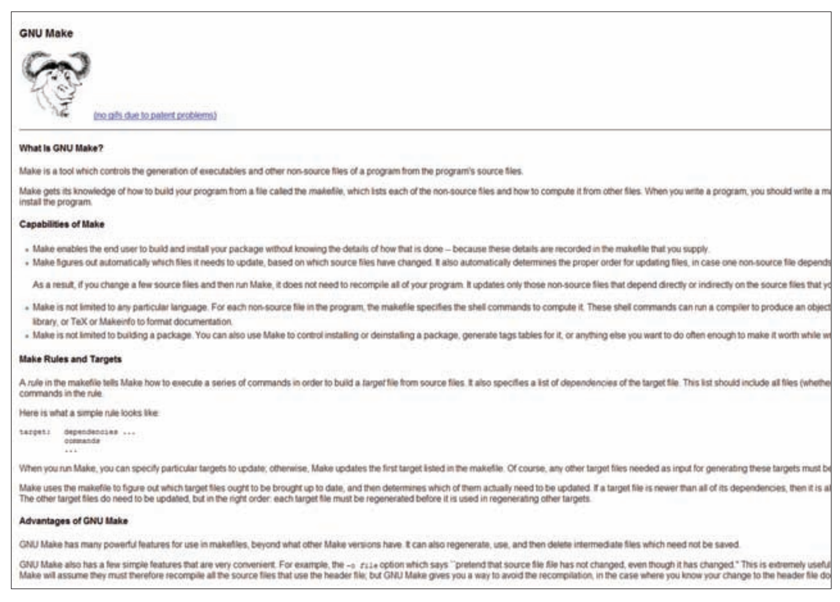
Una herramienta para el administrador de Sistemas

Make fue creado con el objetivo de automatizar compilaciones. Sin embargo también puede ser utilizado para automatizar tareas corrientes de administración de sistemas. Es usual verlo en la compilación de las tablas (mapas) de un sistema de correo postfix. Esto libera al administrador del sistema de correo la tediosa conversión de las tablas al formato binario que utiliza postfix.

Otro ejemplo es la administración centralizada de firewalls en el paquete shorewall-lite. Shorewall es un firewall que parte de una serie de archivos de configuración y genera reglas de iptables, liberando de escribir las reglas a la vez que nos da una interfaz de alto nivel a las reglas de firewall. Shorewall posee una versión para ser instalada en equipos controlados de forma centralizada, el shorewall-lite. Aquí se utiliza un makefile para generar las reglas y subirlas al equipo remoto. En el listado adjunto, obtenido del proyecto podemos ver un ejemplo completo, aplicado en una tarea netamente de administración de sistemas. En este caso el uso de make y ssh con llaves privadas facilita automatizar la administración de una gran cantidad de firewalls.

¿Problemas?

Una fuente de errores común es olvidar el salto de tabulador al comienzo de una línea de comando. Es importante configurar los editores de texto para que no remplacen las marcas de tabulador por espacios.


GNU Make

(no gifs due to patent problems)

What is GNU Make?

Make is a tool which controls the generation of executables and other non-source files of a program from the program's source files.

Make gets its knowledge of how to build your program from a file called the *makefile*, which lists each of the non-source files and how to compute it from other files. When you write a program, you should write a m[...] install the program.

Capabilities of Make

- Make enables the end user to build and install your package without knowing the details of how that is done -- because these details are recorded in the makefile that you supply.
- Make figures out automatically which files it needs to update, based on which source files have changed. It also automatically determines the proper order for updating files, in case one non-source file depends[...]

  As a result, if you change a few source files and then run Make, it does not need to recompile all of your program. It updates only those non-source files that depend directly or indirectly on the source files that yo[...]
- Make is not limited to any particular language. For each non-source file in the program, the makefile specifies the shell commands to compute it. These shell commands can run a compiler to produce an object[...] library, or TeX or Makeinfo to format documentation.
- Make is not limited to building a package. You can also use Make to control installing or deinstalling a package, generate tags tables for it, or anything else you want to do often enough to make it worth while wr[...]

Make Rules and Targets

A *rule* in the makefile tells Make how to execute a series of commands in order to build a *target* file from source files. It also specifies a list of dependencies of the target file. This list should include all files (whethe[...] commands in the rule.

Here is what a simple rule looks like:

```
target:   dependencies ...
          commands
          ...
```

When you run Make, you can specify particular targets to update; otherwise, Make updates the first target listed in the makefile. Of course, any other target files needed as input for generating these targets must be[...]

Make uses the makefile to figure out which target files ought to be brought up to date, and then determines which of them actually need to be updated. If a target file is newer than all of its dependencies, then it is al[...] The other target files do need to be updated, but in the right order: each target file must be regenerated before it is used in regenerating other targets.

Advantages of GNU Make

GNU Make has many powerful features for use in makefiles, beyond what other Make versions have. It can also regenerate, use, and then delete intermediate files which need not be saved.

GNU Make also has a few simple features that are very convenient. For example, the `-o file` option which says "pretend that source file *file* has not changed, even though it has changed." This is extremely useful[...] Make will assume they must therefore recompile all the source files that use the header file; but GNU Make gives you a way to avoid the recompilation, in the case where you know your change to the header file do[...]

Figura 3. Página de GNU Make

Un problema común es la representación de comandos que se extienden en múltiples líneas. Es importante proteger los saltos de línea y utilizar los puntos y comas necesarios. En el siguiente listado dejamos un ejemplo de comandos que se extienden en múltiples líneas:

```
mkdir:
    if [ -z $"{TMPDIR}" ] \
    then \
    mkdir ${TMPDIR} ; \
          fi \
```

Conclusiones

A lo largo del artículo hemos presentado una rápida introducción a make. El lector está capacitado para empezar a aplicarlo en sus proyectos de programación. Quienes estén interesados en profundizar en el uso de make el GNU make cuenta con un completo manual en formato textinfo. Este manual puede resultarle un poco árido al lector, por lo que recomendamos hacer el esfuerzo de leer en inglés el excelente libro de la editorial O'reilly sobre make: "Managing Projects with GNU Make". El mismo está disponible para descarga en: *http://oreilly.com/catalog/make3/book/index.csp*.

Como conclusión dejamos un último ejemplo de makefile, para que el lector ejecute. Un pequeño tributo a la cultura UNIX, que este año cumple sus primeros 40 años de existencia.

love: `@echo not war?`



Sobre el autor

Andrés Tarallo se desempeña como administrador de sistemas en una empresa de contenidos web, íntegramente montada sobre plataformas libres. Su línea de trabajo es desarrollo de aplicaciones WEB, en lenguajes PERL, PHP y JAVA. Trabaja simultáneamente como consultor e integrador de sistemas para compañías pequeñas y medianas en Uruguay, integrando redes heterogéneas o migrándolas a plataformas libres. Ha dado charlas sobre tecnologías basadas en software libre en diversas conferencias en Uruguay, Argentina y Brasil. Estudió en la Universidad ORT Uruguay, donde obtuvo título de Analista Programador.

Programando con inteligencia (artificial)

José María Gómez Hidalgo

Programar no siempre es fácil, y menos aún cuando las aplicaciones a programar son sofisticadas. Éste es el caso de los programas capaces de imitar, de alguna forma, las capacidades humanas de comunicación, razonamiento o aprendizaje, es decir, programas basados en Inteligencia Artificial.



linux@software.com.pl

Sin embargo, la comunidad del software libre se esfuerza en hacernos las cosas más fáciles, poniendo a nuestra disposición bibliotecas que incorporan funciones muy avanzadas. En esta serie de artículos que aquí comenzamos, revisaremos una de las bibliotecas más interesantes por su simplicidad, potencia y versatilidad, que se llama WEKA, y veremos como construir programas muy avanzados con gran rapidez y sencillez.

John McCarthy, uno de los padres de la Inteligencia Artificial, y que además acuñó el término, define esta disciplina como "la ciencia de la producción de máquinas inteligentes, especialmente programas inteligentes". El objetivo planteado es muy ambicioso, especialmente teniendo en cuenta que ni siquiera hay un consenso absoluto en lo que se define como "inteligencia". Sin embargo, y por simplicidad, podemos simplemente decir que pretendemos desarrollar programas con capacidades similares a las humanas en ámbitos como la percepción (visión artificial), el razonamiento (deducción, predicción, etc.), el movimiento (robótica), la comunicación (análisis y generación de lenguaje y habla), o el aprendizaje.

Dentro de este marco tan general, nos vamos a ocupar en especial del tema del Aprendizaje Automático, es decir, el desarrollo de programas capaces de mejorar su efectividad con la experiencia. Este tema es especialmente interesante por su aplicabilidad a todos los ámbitos que se nos ocurran. Por ejemplo, mientras escribo este artículo, OpenOffice está tratando de predecir en todo momento cual es la siguiente palabra que voy a utilizar. Si esta herramienta de predicción es capaz de registrar los éxitos (cuando pulso Intro porque la palabra que está autocompletando es la correcta) y los fallos (cuando ignoro su propuesta), y aprender a partir de ellos, sus predicciones serán cada vez mejores.

La construcción de programas que sean capaces de aprender no es una tarea difícil, y de hecho, numerosos programas incorporan esta capacidad de una manera u otra. El problema reside en que muchos programas incorporan un algoritmo "ad-hoc", es decir, diseñado específicamente para el problema que se trata y probado de una



Figura 1. La interfaz de inicio WEKA, con los botones de acceso a sus cuatro sub-interfaces

manera limitada. Esto es un problema porque existen literalmente decenas de algoritmos de aprendizaje (sólo en la familia de las Redes Neuronales hay múltiples tipos), y no hay garantías de que en un programa concreto se esté utilizando el algoritmo más efectivo, es decir, el que garantiza mayor eficacia y rapidez.

Para la experimentación con algoritmos de aprendizaje, tanto desde el punto de vista más científico como desde el más pragmático, han surgido en los últimos años múltiples bibliotecas software que incorporan no sólo muchos de los algoritmos preprogramados, sino además un entorno completo para evaluarlos tanto a nivel de eficacia como de rendimiento. Usando uno de estos entornos, el programador puede efectuar gran cantidad de pruebas que le permiten escoger el algoritmo más eficaz para su problema, con garantías de que será realmente útil en su aplicación. Algunos de estos entornos son:

- WEKA – Un entorno de experimentación con licencia libre escrito en Java por el equipo del Profesor Ian Witten y Eibe Frank, de la Universidad de Waikato en Nueva Zelanda.
- Orange – Una biblioteca para el aprendizaje automático, basada en componentes y escrita en Python, desarrollada con licencia libre por el laboratorio de Inteligencia Artificial de la Universidad de Liubliana en Eslovenia.
- RapidMiner (Community Edition) – Un entorno en Java que incluye WEKA y que permite definir los experimentos en XML, desarrollado originalmente por la Unidad de Inteligencia Artificial de la Universidad de Dormund, y explotado comercialmente por Rapid-I.
- Proyecto R – Un entorno para el cómputo estadístico y la visualización de datos y gráficas escrito en C y C++ por John Chambers y otros en los laboratorios Bell de Lucent Technologies, y distribuido con licencia GNU.

Una característica adicional de todos estos entornos es que ofrecen una API de programación, es decir, que pueden ser llamados desde otros programas, muchas veces incluso en lenguajes diferentes. Por ello, tiene mucho sentido desarrollar un prototipo de un sistema, acumular datos y experimentar con el paquete elegido, y una vez determinado el algoritmo más adecuado y efectivo, incorporarlo en el sistema con unas cuantas llamadas.

De entre todas estas herramientas y otras existentes, nosotros hemos seleccionado WEKA por varias razones:

- Se trata de uno de los entornos más populares, utilizado en numerosos trabajos de investigación y en múltiples entornos comerciales.
- La interfaz es simple e intuitiva (una vez que se sabe de que se trata el Aprendizaje Automático).
- Incluye una de las colecciones más extensas de algoritmos del mercado, y permite experimentar con Redes Neuronales,

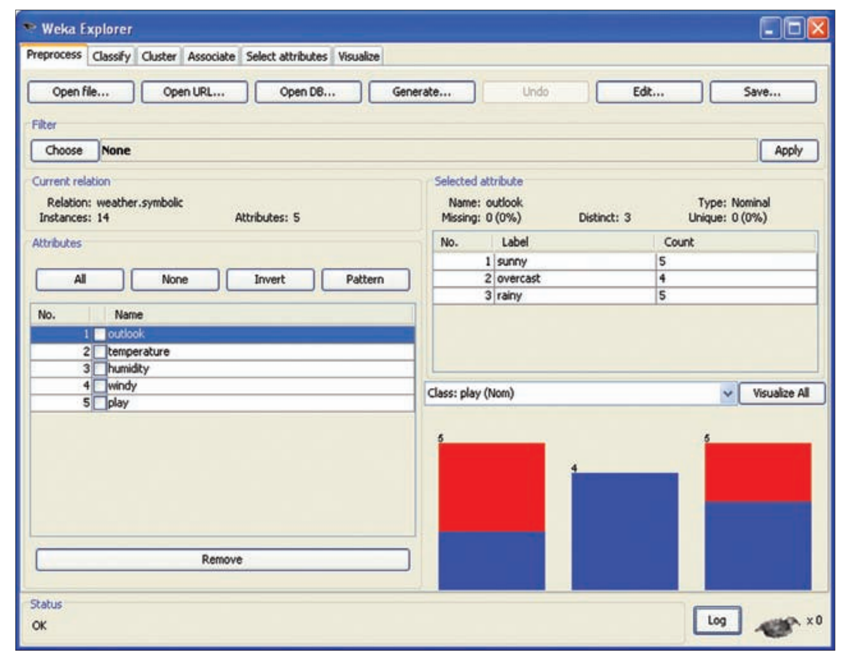


Figura 2. La interfaz del *Explorer* de WEKA, en la pestaña *Preprocess* con datos cargados

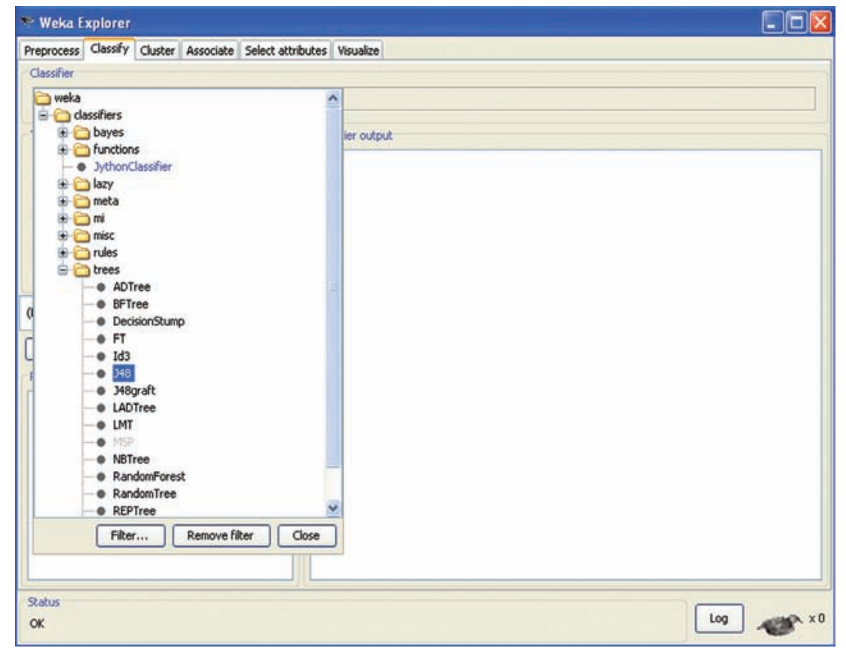


Figura 3. Selección del clasificador J48 en la pestaña *Classify* del *Explorer*

algoritmos basados en reglas o árboles de decisión, modelos bayesianos y probabilísticos como las redes de Inferencia Bayesianas, etc.

- La API de programación es extremadamente simple, y con dos o tres llamadas, es posible integrar cualquier algoritmo en un programa.
- La mayoría de las veces, trabajar en Java no es un problema porque en servidor (Servlets, JSPs) es tan rápido y eficiente como muchos otros lenguajes.

En nuestra propia experiencia, WEKA es un entorno potente y simple que permite el desarrollo rápido de programas con capacidades inteligentes. Como además es muy intuitivo, nos permite evitar tener que exponer los conceptos básicos sobre el Aprendizaje Automático, que iremos presentando al mismo tiempo que la herramienta.

Instalación de WEKA

La última versión estable de WEKA es la 3.6, y está disponible para su descarga en el apartado "Download" de la página web. Al ser un software independiente de la plataforma, es totalmente portable entre sistemas operativos, y existen tres versiones principales: para Windows, Mac OS X y Linux. En el caso de Windows existen dos versiones del instalador, una que contiene la Máquina Virtual de Java (el Java Runtime Environment, v. 1.5), y otra sin ella. Para Mac OS X, se distribuye una imagen de disco, y para Linux el archivo ZIP (`weka-3-6-1.zip`) que contiene el código ejecutable, `weka.jar`. En el caso de este último sistema, el programa se lanza con la orden:

```
java -jar weka.jar
```

Obviamente se precisa Java (v. 1.5 o superior), y al lanzar el programa de esta manera, se anula la variable `$CLASSPATH` del entorno. En el caso de Windows, la herramienta se puede lanzar desde el menú de inicio.

Primera prueba

La primera prueba que realizaremos nos va a indicar ya los primeros pasos del concepto de aprendizaje supervisado de un clasificador, que es uno de los de los dos métodos básicos de clasificación en el Aprendizaje Automático. En el aprendizaje supervisado, se toma un conjunto de datos etiquetados y un algoritmo de aprendizaje, y se crea un modelo de clasificación o clasificador, que se puede usar

Listado 1. Salida del clasificador J48 sobre la colección `weather.nominal.arff`

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    evaluate on training data
=== Classifier model (full training set) ===
J48 pruned tree

------------------

outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)
Number of Leaves  :        5
Size of the tree :         8
Time taken to build model: 0 seconds

=== Evaluation on training set ===

=== Summary ===

Correctly Classified Instances          14               100 %
Incorrectly Classified Instances         0                 0 %
Kappa statistic                                            1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0 %
Root relative squared error              0 %
Total Number of Instances               14
=== Detailed Accuracy By Class ===
                       TP Rate   FP Rate   Precision   Recall   F-
Measure    ROC Area  Class
1          0                     1         1                1
1                yes
                                 1         0                1
1          1                     1        no
Weighted Avg.                    1         0                1
                                 1         1                1
=== Confusion Matrix ===
 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no
```

para clasificar nuevos datos si etiquetar. La etiqueta es la clase, y en su versión más sencilla, la clasificación es binaria: me gusta o no me gusta una página Web, un correo es spam o no, etc. Partiendo de datos ya clasificados (una serie de páginas Web que me gustan y otras que no, una serie de mensajes basura y otros que no lo son, etc.), el algoritmo analizará las razones de porque clasificar los datos en una clase u otra, y almacenará esta información. Cuando lleguen nuevos datos sobre los que no se sabe nada (una nueva página Web, un mensaje de correo, etc.), usará esa información almacenada para predecir la clase (es decir, nos dirá si nos va a gustar o no, o si el correo es basura o no, etc.).

La interfaz que muestra el sistema WEKA cuando se arranca se muestra en la figura 1. En esta interfaz, existen una serie de menús que ignoraremos de momento, y cuatro botones que dan acceso a las cuatro interfaces de WEKA. De las cuatro, la que usaremos nosotros es la interfaz llamada *Explorer* (explorador). Pulsando sobre ella, aparece la interfaz que se muestra en la figura 2.

La interfaz *Explorer* está basada en pestañas, que aparen en la parte superior. El arranque se realiza sobre la pestaña *Preprocess* (preproceso), destinada a la carga y formateado de datos. El resto de pestañas tienen distintas aplicaciones, y las iremos revisando a medida que avancemos en el trabajo. En realidad, la interfaz de la figura 2 no aparece como tal en el inicio, sino que hemos realizado ya una primera carga de datos para que el sistema aprenda de ellos. Esta carga se realiza pulsando sobre el botón *Open file...* (abrir archivo), seleccionando en la ventana de exploración de disco que aparece el archivo `weather.nominal.arff` situado en el directorio `data` del lugar donde hayamos instalado WEKA (en Linux, la carpeta donde descomprimimos el archivo ZIP, en Windows típicamente en `C:\Archivos de programa\Weka-3-6`), y aceptando.

Para la primera prueba, vamos a pasar a la pestaña *Classify* (clasificar), que se usa para invocar algoritmos de aprendizaje sobre los datos que hemos cargado previamente. En el botón *Choose* (elegir) del área Classifier (clasificador) de la parte superior de esta ventana, se pulsa y selecciona en el árbol que aparece el algoritmo J48 de la carpeta *trees* (árboles), como se muestra en la figura 3. En la sección *Test options* (opciones de evaluación) de la parte superior izquierda, marcamos la opción *Use training set*

Listado 2. El contenido del archivo `contact-lenses.arff`

```
@relation contact-lenses
@attribute age                    {young, pre-presbyopic, presbyopic}
@attribute spectacle-prescrip     {myope, hypermetrope}
@attribute astigmatism            {no, yes}
@attribute tear-prod-rate {reduced, normal}
@attribute contact-lenses {soft, hard, none}
@data
%
% 24 instances
%
young,myope,no,reduced,none
young,myope,no,normal,soft
young,myope,yes,reduced,none
young,myope,yes,normal,hard
young,hypermetrope,no,reduced,none
young,hypermetrope,no,normal,soft
young,hypermetrope,yes,reduced,none
young,hypermetrope,yes,normal,hard
pre-presbyopic,myope,no,reduced,none
pre-presbyopic,myope,no,normal,soft
pre-presbyopic,myope,yes,reduced,none
pre-presbyopic,myope,yes,normal,hard
...
```

Listado 3. Reglas generadas por el algoritmo PART para los datos de `contact-lenses.arff`

```
PART decision list
------------------
tear-prod-rate = reduced: none (12.0)
astigmatism = no: soft (6.0/1.0)
spectacle-prescrip = myope: hard (3.0)
: none (3.0/1.0)
Number of Rules  :        4
```

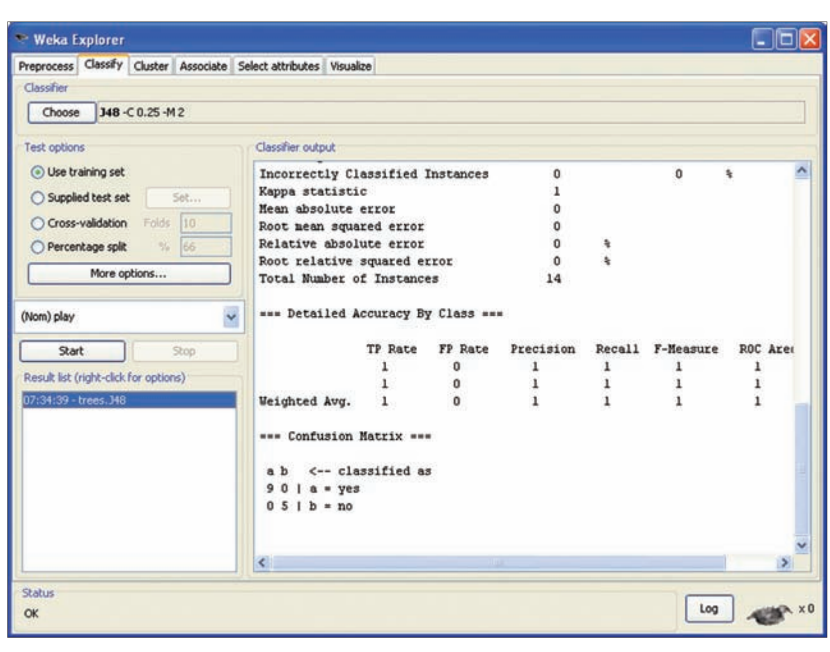


Figura 4. Resultados de la ejecución del clasificador J48 en la pestaña *Classify* del *Explorer*

(usar colección de entrenamiento), y pulsamos el botón Start (comenzar) que aparece más o menos a la mitad del lado izquierdo. Como resultado, el algoritmo J48 se activa sobre los datos cargados y nos muestra los resultados del aprendizaje en la sección *Classifier output* (salida del clasificador) en el lado derecho, como se muestra en la figura 4.

Aun vamos a llegar un poco más lejos. Pulsado con el botón izquierdo en el panel inferior izquierdo, llamado *Result list* (lista de resultados) sobre la entrada `trees.J48`, aparece un menú contextual en el que seleccionamos la opción *Visualize tree* (visualizar árbol), y como resultado, aparece una nueva ventana titulada *Weka Classifier Tree Visualizer* (visualizador de árboles de clasificación de WEKA), donde se muestra el árbol de decisión generado por el algoritmo J48 sobre los datos cargados.

Listado 4. Código de ejemplo de un programa que usar WEKA como biblioteca software

```
import weka.core.Instances;
import weka.core.Instance;
import weka.core.Attribute;
import weka.core.converters.ConverterUtils.DataSource;
import weka.classifiers.rules.PART;
public class EjemploWeka {
public static void main(String[] arg) throws Exception {
 // Leer todos los ejemplares de archivo con la colección
 // de entrenamiento
 DataSource archivo = new DataSource(arg[0]);
 Instances coleccion = archivo.getDataSet();
 // Asignar el último atributo como clase objetivo
 coleccion.setClassIndex(coleccion.numAttributes() - 1);
 // Mostrar por pantalla la colección de entrenamiento
 System.out.println("\nColeccion de entrenamiento:\n");
 System.out.println(coleccion );
 // -----
 // Inicializar y entrenar el clasificador
 PART clasificador = new PART();
 clasificador.buildClassifier(coleccion);
 // Mostrar por pantalla el modelo construido
 System.out.println("\nClasificador PART:\n");
 System.out.println(clasificador);
 // -----
 // Crear un nuevo ejemplo con los valores de la linea
 // de órdenes
 Instance ejemplo = new Instance(5);
 // Marcar la anterior como colección de referencia
 ejemplo.setDataset(coleccion);
 ejemplo.setValue(0,arg[1]);
 ejemplo.setValue(1,arg[2]);
 ejemplo.setValue(2,arg[3]);
 ejemplo.setValue(3,arg[4]);
 // Mostrar el ejemplar por pantalla
 System.out.println("\nEjemplo leído:\n");
 System.out.println(ejemplo);
 // -----
 // Clasificar el nuevo ejemplar
 double indiceClase =  clasificador.classifyInstance(ejemplo);
 // Mostrar la clase del ejemplar por pantalla
 System.out.println("\nClase predecida:\n");
 System.out.println(coleccion.classAttribute().value((int)
indiceClase));
 }
}
```

Explicación

Todo muy visual, pero... ¿qué hemos hecho?, ¿para qué sirve?... Esto merece una explicación detallada. El proceso que hemos realizado para obtener el árbol de decisión de la figura 5 consta de los siguientes pasos.

Carga de datos

Hemos seleccionado el archivo `weather.nominal.arff`, que contiene una colección de datos sobre el problema de decidir si jugar al tenis o no en función del tiempo meteorológico. Por tanto, el objetivo del aprendizaje es, dada una serie de ejemplos en los que en determinadas condiciones meteorológicas, hemos decidido jugar o no, el sistema debe aprender a aconsejarnos sobre si jugar o no cuando introduzcamos nosotros, por ejemplo, las condiciones de hoy. Los datos se caracterizan en función de las variables siguientes:

- *Outlook* (aspecto), que puede valer *sunny* (soleado), *overcast* (despejado) o *rainy* (lluvioso).
- *Temperature* (temperatura), que puede valer *hot* (cálido), *mild* (intermedio) o *cold* (frio).
- *Humidity* (humedad), que puede valer *high* (alta) o *normal* (normal).
- *Windy* (ventoso), que puede valer *TRUE* (cierto) o *FALSE* (falso). Es por tanto una variable *booleana*.
- *Play* (jugar), que puede valer *yes* (sí) o *no* (no). Esta variable es la clase a predecir en función de los datos anteriores. Por ejemplo, si el tiempo es ventoso, mejor no jugar al tenis.

Esta colección consta de catorce ejemplos, que constituyen la colección de entrenamiento o aprendizaje, llamada así porque es sobre la que se aprende.

Selección y ejecución del algoritmo de aprendizaje

Hemos seleccionado el algoritmo J48, que es un clon del algoritmo C4.5, uno de los algoritmos de aprendizaje de árboles de decisión más populares y efectivos. Un árbol de decisión es una estructura de árbol donde se efectúan tests (comprobaciones de valor sobre los atributos) en los nodos internos del árbol, y en función de los resultados obtenidos, se elige un camino u otro entrando por la raíz del



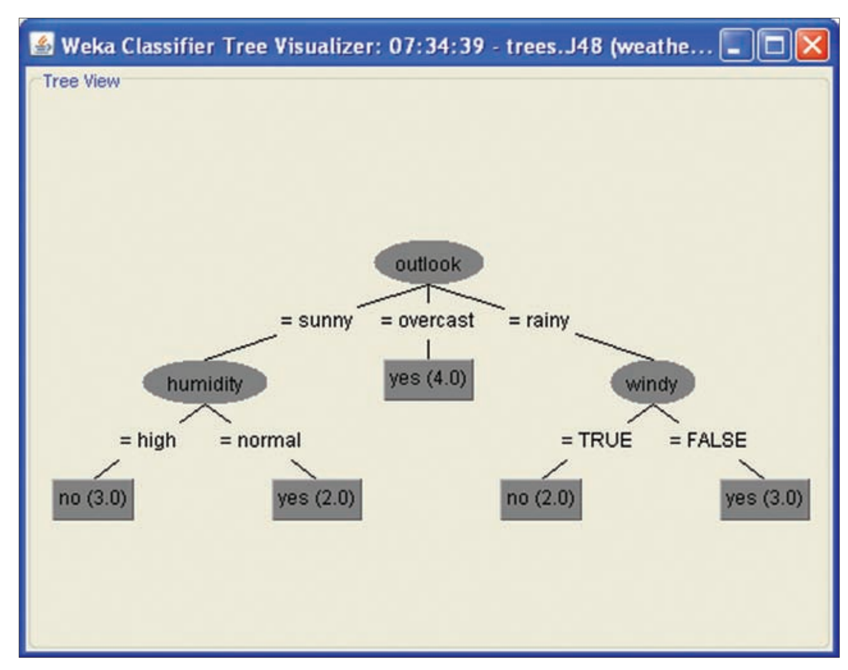


Figura 5. Árbol de decisión obtenido por el clasificador J48 en la pestaña *Classify* del *Explorer*

árbol. Cuando se llega a una hoja, es decir, un nodo sin hijos o ramas descendentes, hay una decisión. En nuestro caso, al árbol de decisión obtenido por este algoritmo es el que se muestra de manera gráfica en la figura 5. Por ejemplo, si entremos por la raíz en la parte superior, lo primero que se pregunta es el valor del atributo *outlook* (aspecto); si en nuestro caso es *overcast* (despejado), llegamos directamente a una hoja etiquetada como yes (sí), luego la decisión es jugar, puesto que este el valor predecido para la variable *play* (jugar). El número que aparece en la hoja (4.0) es el número de ejemplos o ejemplares de la colección de entrenamiento que cumplen la serie de condiciones que se han ido fijando al recorrer el árbol para llegar hasta la hoja. En otras palabras, hay cuatro ejemplos de los catorce en los que el valor de la variable *outlook* es *overcast*, y en todos ellos, la variable *play* vale *yes*.

Evaluación del algoritmo de aprendizaje

La esencia del aprendizaje es que los sistemas mejoren con el tiempo, por lo que la evaluación de los resultados es una componente esencial. Si no se mide, no se progresa.

Los sistemas de clasificación se miden generalmente en términos de efectividad, es decir, de la capacidad de acertar que tiene el clasificador. Para poder medir la efectividad, es preciso disponer de otro conjunto de datos ya clasificados, la colección de evaluación, para comparar lo que decide el clasificador con la clasificación correcta. En este caso, hemos seleccionado los mismos datos de entrenamiento como colección de evaluación en *Test options* (opciones de evaluación), con los resultados que se presentan en la figura 4, en el área de *Classifier output* (salida del clasificador). La salida completa del clasificador J48 sobre la colección `weather.nominal.arff` se muestra en el listado 1.

La salida de un clasificador en WEKA se divide en tres secciones:

- *Run information* (información de la ejecución) – indica los parámetros del clasificador utilizado y las características de la colección de datos.
- *Classifier model* (modelo de clasificación) – presenta el modelo construido por el algoritmo. En nuestro caso, se trata de un árbol de decisión que se muestra en modo texto con sangrados.
- *Evaluation on training set* (evaluación en el conjunto de entrenamiento) – en general, se trata de la sección dedicada a la evaluación. Se presentan una serie de medidas de eficacia, de entre las cuales las más simples de entender son las dos primeras: *Correctly Classified Instances* (ejemplares correctamente clasificados), e *Incorrectly Classified Instances* (ejemplares incorrectamente clasificados). Ade más, se muestra la tabla de confusión o tabla de contingencia (*Confusion Matrix*), que indica los números de aciertos y de errores.

La tabla de contingencia es muy importante, y conviene explicarla con detalle. Se trata de

Listado 5. Salida del programa de ejemplo

```
Colección de entrenamiento:
@relation contact-lenses
@attribute age {young,pre-presbyopic,presbyopic}
@attribute spectacle-prescrip {myope,hypermetrope}
@attribute astigmatism {no,yes}
@attribute tear-prod-rate {reduced,normal}
@attribute contact-lenses {soft,hard,none}
@data
young,myope,no,reduced,none
young,myope,no,normal,soft
young,myope,yes,reduced,none
young,myope,yes,normal,hard
...
presbyopic,hypermetrope,yes,normal,none
Clasificador PART:
PART decision list
------------------
tear-prod-rate = reduced: none (12.0)
astigmatism = no: soft (6.0/1.0)
spectacle-prescrip = myope: hard (3.0)
: none (3.0/1.0)
Number of Rules  :      4
Ejemplo leído:
young,myope,no,reduced,?
Clase predecida:
none
```

una tabla de doble entrada, de modo que en las columnas se muestran los resultados del clasificador (*classified as*, clasificados como) y en las filas los resultados correctos. En el caso de nuestro problema del consejero del juego de tenis, hay dos clases: *yes* (sí) y *no* (no). En la casilla que corresponde a las coordenadas (a,a), es decir, (*yes*,*yes*), aparece en número de ejemplos de la clase a (*yes*) clasificados por J45 en la clase a (*yes*), y por tanto son aciertos (usualmente llamados *true positives*, positivos ciertos). Lo mismo ocurre en la entrada (b,b), pero con la clase b (*no*); son los llamados positivos falsos (*false positives*). En las casillas (a,b) y (b,a) se trata de errores del sistema, bien porque ha clasificado como b (*no*) cuando es a (*yes*), como a la inversa. En el primer caso se trata de falsos negativos (*false negatives*), y en el segundo de falsos positivos (*false positives*).

Esta tabla proporciona la base de los cálculos de eficacia. Por ejemplo, la precisión (*accuracy*, o porcentaje de aciertos) se calcula sumando todos los aciertos y dividiendo por el número de ejemplares de evaluación (es decir, la suma de las cuatro casillas).

Los atributos y los archivos

Nuestros siguientes pasos serán explicar con detalle los procesos básicos del aprendizaje en base a la interfaz de WEKA, empezando por la selección y caracterización de los datos. Para ello, tomaremos otro ejemplo ilustrativo, el archivo `contact-lenses.arff`, que se encuentra disponible en el directorio `data` dentro del directorio de WEKA. En el listado 2 se muestra una parte del contenido de este archivo, del que hemos eliminado los comentarios iniciales y algunos ejemplares.

En este ejemplo se trata el problema de la recomendación de lentes de contacto en función de las características oculares de un eventual paciente. El archivo, como todos los archivos de WEKA, es de tipo ARFF (*Attribute-Relation Format File*, archivo de formato de relación de atributos), y en realidad es una tabla de base de datos, que consta de tres secciones.

En primer lugar, la sección `@relation`, que contiene el nombre de la relación o tabla. En nuestro caso, la tabla se llama *contact-lenses* (lentes de contacto).

A continuación, la sección de atributos, denotada por `@attribute` (atributo) para cada variable. Las variables se suelen llamar atributos o características en el campo del Aprendizaje Automático. Una de las mayores dificultades en este campo reside en averiguar cuales son los mejores atributos para caracterizar un determinado problema, y normalmente requiere de la experiencia de un experto. En este caso, sobre la base del conocimiento de un oftalmólogo, se sabe que los atributos de un paciente a tener en cuenta para seleccionar el tipo de lentes más adecuadas son:

- La edad (*age*), que puede valer *young* (joven), *pre-presbyopic* (pre-presbiópico) y *presbyopic* (presbiópico). La presbiopía se llama también presbicia o vista cansada.
- La prescripción correctiva (*spectacle-prescrip*) hace referencia al defecto ocular a corregir, y puede valer *myope* (miope) o *hypermetrope* (hipermétrope).
- El astigmatismo (*astigmatism*) es un defecto complementario a los anteriores, y puede valer *no* (no) o *yes* (sí).
- La tasa lacrimal (*tear-prod-rate*) es la cantidad de lágrima que tiene el paciente, y puede ser *reduced* (reducida) o *normal* (normal).
- El atributo *contact-lenses* (lentes de contacto) es la recomendación médica que se da en función de los factores anteriores, y puede ser *soft* (lentes blandas), *hard* (lentes duras) o *none* (ningún tipo de lente de contacto).

Para finalizar, la sección `@data` lista los datos en formato de un ejemplo por línea, y en cada línea los valores separados por una coma.

Por costumbre, se suele poner el atributo a predecir el último, y es la clase objetivo. Este el el valor por defecto en WEKA, aunque se puede seleccionar cualquier otro como clase. Una característica especial de esta colección de datos es que la clase no es binaria, sino ternaria, es decir, puede tomar tres valores.

El resultado de seleccionar el archivo `contact-lenses.arff` en la pestaña *Preprocess* (preproceso) del *Explorer* de WEKA al pulsar sobre *Open file...* (abrir archivo) se muestra en la figura 6. Esta pestaña consta de las siguientes secciones:

- El área superior contiene los botones de acción, que permiten abrir un archivo, una URL (*Open URL...*), una base de datos (*Open DB...*), generar una serie de datos pseudo-aleatorios según una serie de algoritmos disponibles (*Generate...*), deshacer (*Undo*), editar los datos actuales (*Edit...*), o guardar los datos actuales (*Save...*).
- El área *Filter* (filtro), inmediatamente debajo, permite configurar un filtro para la eliminación, cambio o transformación de los datos que cumplan determinadas condiciones.
- Justo debajo a la izquierda, se muestran los datos básicos de la colección actual en el área *Current relation* (relación actual). Estos datos son el nombre de la relación (*Relation*) número de atributos (*Attributes*, cinco en nuestro caso) y el de ejemplares (*Instances*, veinticuatro en nuestro caso).

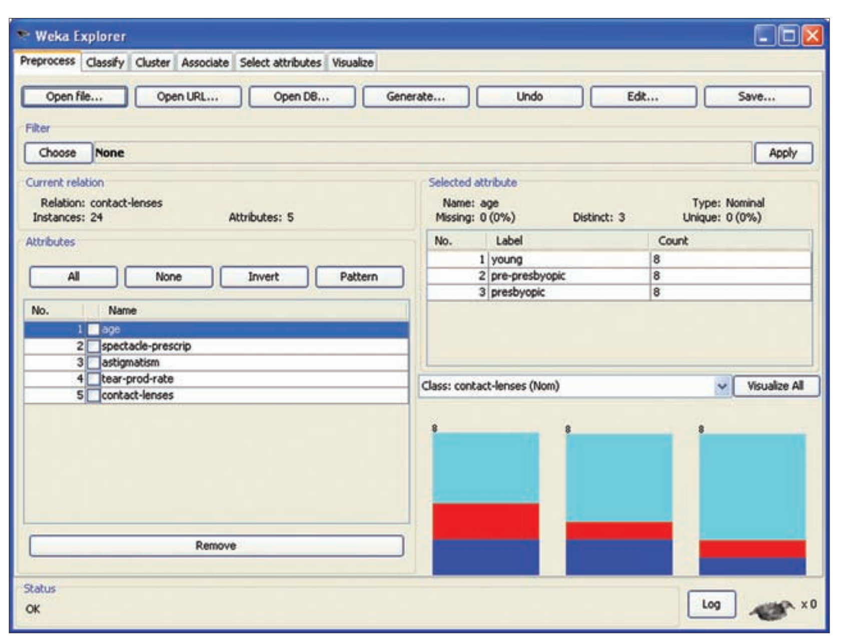


Figura 6. El archivo `contact-lenses.arff` cargado en la pestaña *Preprocess* del *Explorer*



- El área *Attributes* (atributos), más abajo a la izquierda, muestra un listado manipulable de los atributos disponibles en la colección actual. Las acciones disponibles se activan por medio de los botones, que incluyen *All* (todos, seleccionar todos), *None* (ninguno, no seleccionar ninguno), *Invert* (invertir, seleccionar el complementario de la selección actual, *Pattern* (patrón, seleccionar de acuerdo a una expresión regular en Perl que se escribe en una ventana emergente), y *Remove* (eliminar, elimina los atributos seleccionados).
- En el área superior de la derecha (*Selected attribute*, atributo seleccionado), se muestran los posibles valores del atributo seleccionado actualmente (en la figura 6 son los del atributo *age*, edad), junto con una serie de datos sobre el mismo, que incluyen su nombre (*Name*), su tipo (*Type*, en nuestro caso es *nominal*, nominal), el número y porcentaje de ejemplos que carecen de un valor para este atributo (*Missing*, sin valor), el número de valores distintos (*Distinct*, distintos), y el número y porcentaje de valores que aparecen en un sólo ejemplo (*Unique*, únicos).
- El área inferior de la derecha muestra unas barras de colores que indican cuantos ejemplares de cada clase tienen cada posible valor del atributo. Por ejemplo, la barra de la derecha indica que los ocho ejemplares que tiene el valor *presbyopic* (presbiótico) se distribuyen a razón de seis en la clase *none* (ninguna, en azul claro), y uno en cada una de las clases *hard* (duras, en color rojo) y *soft* (blandas, en azul marino). Cuando un atributo muestra un valor para el cual la columna es prácticamente de un solo color, indica que predice con gran claridad ese valor de la clase.
- El área inferior Status (estado) muestra el estado de la herramienta, y a la derecha se muestra un botón Log (registro) que abre una ventana emergente que lista las acciones realizadas hasta el momento, y el símbolo de WEKA acompañado de un número que indica el número de pasos restantes en la acción en curso. El símbolo de WEKA es un pájaro neozelandés que se anima cuando el sistema está realizando alguna acción.

Los atributos que se utilizan para caracterizar un problema de clasificación determinan la calidad del aprendizaje de manera crítica, por lo que esta pantalla permite numerosas opciones de prueba y análisis de los mismos.

Los clasificadores de WEKA

La componente más útil de WEKA es el número de algoritmos de aprendizaje que incluye, accesibles en la pestaña *Classify* (clasificar). En la versión actual (v. 3.6.1) se incluyen:

- Trece algoritmos bayesianos (*bayes*), que generan tablas de probabilidades a partir de las cuales es posible calcular la probabilidad de cada clase objetivo.
- Diecisiete funciones de clasificación (*functions*), que obtienen distintas funciones que computan el valor de la clase a partir de una serie de parámetros dependientes del algoritmo. Dentro de esta familia se incluyen las redes neuronales, que en WEKA son, por ejemplo, *MultilayerPerceptron* (perceptrón multicapa) o *VotedPerceptron* (perceptrón con votos).
- Cinco algoritmos perezosos (*lazy*), que no generan un modelo sino que comparan el ejemplar a clasificar con los de entrenamiento utilizando una función de similitud, y en función de los ejemplares más similares, toman la decisión de clasificación.
- Treinta y cinco algoritmos de comités (*meta*), que se utilizan para combinar la salida de otros algoritmos de aprendizaje. Por ejemplo, el algoritmo *AdaBoost* ejecuta múltiples veces un algoritmo subyacente (de cualquiera de los demás tipos) y combina los resultados para aumentar la efectividad del algoritmo subyacente de manera espectacular.
- Catorce algoritmos para aprendizaje multiinstancia (*MI*), a aplicar cuando la clase objetivo tiene etiquetas definidas de manera imprecisa.
- Once algoritmos de generación de reglas (*rules*), que construyen reglas de clasificación similares a las de un sistema experto.
- Dieciséis algoritmos de generación de árboles de decisión (*trees*).
- Cinco algoritmos clasificados como miscelánea (*misc*), no pertenecientes a ninguna de las familias anteriores.

Con un total de ciento dieciséis algoritmos pertenecientes a prácticamente todas las grandes familias de algoritmos de aprendizaje, y cada uno de ellos parametrizable de múltiples formas, la potencia experimental de la biblioteca software WEKA no tiene precedentes, ni siquiera en el campo del software comercial.

El uso y evaluación de clasificadores

Para explicar el uso y evaluación de clasificadores, seguiremos con el ejemplo de `contact-lenses.arff`. En la pestaña *Classify* (clasificar), seleccionamos el algoritmo PART (`weka.classifier.rules.PART`, accesible al pulsar el botón *Choose* – elegir, del área *Classifier* – clasificador), seleccionamos la opción de validación cruzada (*Cross-validation*) con diez carpetas (*Folds*) en el área *Test options* (opciones de evaluación), y pulsamos el botón *Start* (ini-

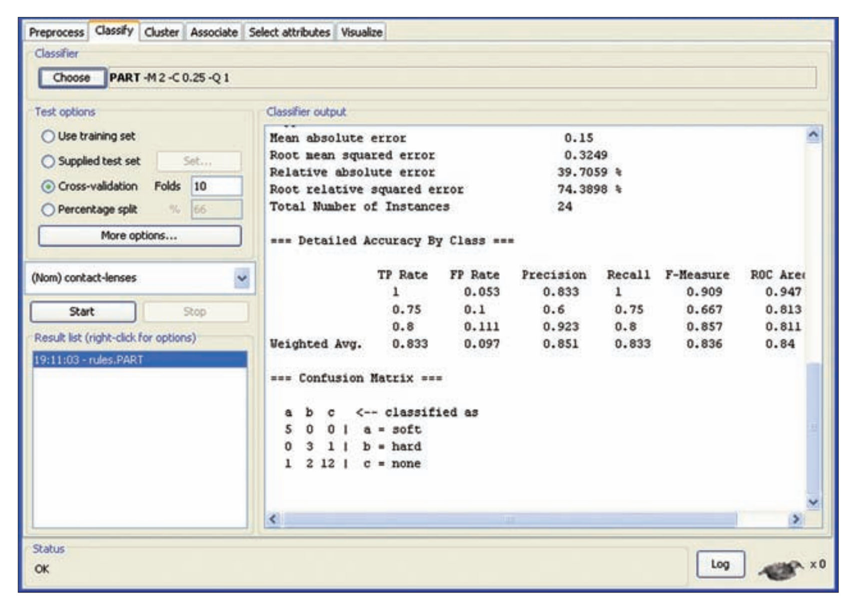


Figura 7. El resultado de ejecutar PART sobre el archivo `contact-lenses.arff`

ciar). De esta manera obtenemos el estado que se muestra en la figura 7.

Los clasificadores a utilizar se seleccionan en el botón *Choose* (elegir). Dentro del amplio rango de clasificadores de WEKA, hemos seleccionado PART, que es un clasificador basado en reglas, es decir, que dados unos datos de entrenamiento, genera un conjunto de reglas de la forma "if-then", es decir, "si-entonces". A partir de condiciones sobre los atributos y sus valores, las reglas definen la clase que hay que asignar. Las reglas generadas por el clasificador PART son las que aparecen en el listado 3.

Las reglas generalmente se aplican ordenadas secuencialmente, es decir, en primer lugar se observa la primera, y si la condición no se cumple, la segunda, y así sucesivamente. En este caso, la primera regla indica que en caso de que el valor del atributo *tear-prod-rate* (tasa lacrimal) sea *reduced* (reducida), el valor de la clase debe ser *none* (ninguna). El número que acompaña a esta regla es doce, que indica que hay doce ejemplos de la colección de entrenamiento que la cumplen (en los que el atributo tiene ese valor), y todos ellos están en la clase *none* (ninguna). La última regla no tiene condición previa (el lado izquierdo de los dos puntos está vacío), por lo que es la regla por defecto o regla a aplicar cuando las demás fallan, y la clase asignada también es *none* (ninguna), pero hay dos números: el número 3.0 indica que hay tres ejemplos que no cumplen ninguna de las condiciones de aplicación de las reglas anteriores, pero tiene la clase *none* (ninguna), mientras que el número 1.0 indica que hay un ejemplo que tampoco cumple las condiciones anteriores, pero su clase no es *none* (ninguna).

Cuando se selecciona un algoritmo, es posible modificar sus parámetros pulsando sobre el campo donde aparece su nombre en la parte superior. Si se pulsa en el caso de PART, aparece la ventana de la figura 8, en la que además hemos pulsado sobre el botón *More* (más), para obtener la ventana de la derecha que nos muestra un texto explicativo sobre el algoritmo.

La tabla de confusión, que aparece en el área *Classifier output* (salida del clasificador) a la derecha, tiene en este caso nueve casillas, de las cuales, las tres que se encuentran en la diagonal principal corresponden a aciertos (que suman veinte), y el resto de casillas se corresponden con errores. El método de evaluación utilizado es más sensato que en la ocasión anterior, ya que antes hemos evaluado el clasificador sobre la colección de entrenamiento (*Use training set*, usar colección de entrenamiento), y es lógico que la eficacia de un clasificador sobre los datos sobre los que ha aprendido sea muy alta. Hay tres métodos más de evaluación:

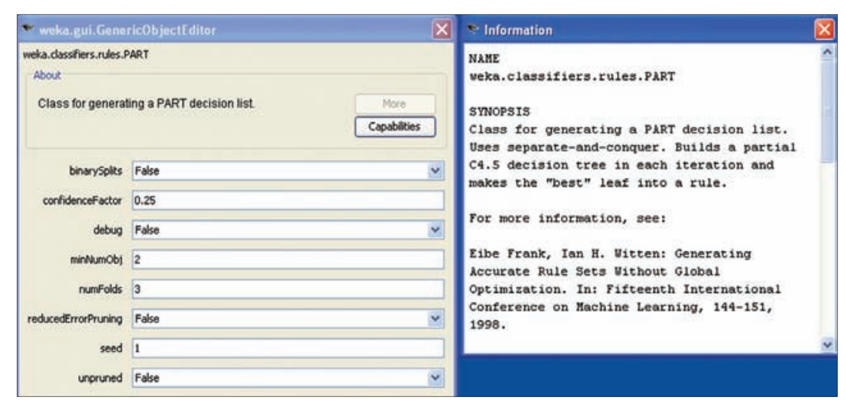


Figura 8. Ventana para editar los parámetros de PART y más información sobre él

- *Supplied test set* (colección de evaluación adicional), que permite usar otro archivo con la misma estructura (los mismos atributos en el mismo orden, y la clase de cada uno de los ejemplos que en él aparezcan) para la evaluación. Es preciso conocer la clase porque si no, no es posible saber si cada intento se trata de un acierto o un fallo.
- *Cross-validation* (validación cruzada), que consiste en dividir los datos aleatoriamente en N grupos (en nuestro ejemplo, en 10), de los cuales se usa por turnos N-1 para entrenar, y 1 para clasificar y evaluar. Los resultados se acumulan en una sola tabla.
- *Percentage split* (división por porcentaje), que consiste en dividir los datos en dos partes según el porcentaje seleccionado. La primera parte (el porcentaje seleccionado) se usa para entrenamiento, y la segunda para clasificación.

Estos métodos utilizan todos ellos datos nuevos para la evaluación, por lo que la efectividad del algoritmo se va a resentir. Esto es lo razonable, puesto que no cabe esperar más que en contadas ocasiones, los clasificadores entrenados acierten siempre.

Un programa que usa WEKA

El objetivo de este artículo no es usar WEKA desde su interfaz, sino desde otro programa. Aparte de la interfaz, donde hemos visto sólo una de las cuatro posibles, existen otros dos modos de utilizar WEKA:

- Desde la linea de órdenes, por medio de llamadas a los algoritmos concretos sobre los datos deseados.
- Desde un programa, con llamadas a las funciones disponibles en la biblioteca software.

La primera de estas dos formas es equivalente a utilizar la interfaz *Simple CLI* (*Simple Command Line Interface*, interfaz simple en linea de órdenes), disponible en el arranque. La segunda la vamos a examinar con detalle sobre el ejemplo del listado 4.

El ejemplo, que se debe guardar en el archivo `EjemploWeka.java`, se ha mantenido expresamente lo más simple posible, por lo que no se definen clases sino un solo programa principal (`main`), y la lectura del nombre del archivo de datos de entrenamiento y de los datos del nuevo ejemplar a clasificar se realiza desde la linea de órdenes (`String[] arg`) sin control de los errores de lectura (`throws Exception`, sin ninguna sección `try-catch`). Este ejemplo consta de las siguientes secciones:

- La sección de importación, que incluye los objetos que son necesarios para albergar un conjunto de ejemplares de entrenamiento (`Intances`), el nuevo ejemplo a clasificar (`Instance`), el tipo de los atributos (`Attribute`), el objeto para cargar los datos de entrenamiento de un archivo (`DataSource`) y el clasificador PART (`PART`).
- La sección de lectura de los datos de entrenamiento, que carga el archivo que aparece como primer argumento de la línea de órdenes (`arg[0]`) por medio de un objeto `DataSource`, y se lo asigna a un objeto de colección de ejemplares (`Instances`) usando el método `getDataSet()` del objeto de tipo `DataSource`. Además establece como clase el último



atributo de cada ejemplar usando los métodos `setClassIndex()` y `num Attributes()` del objeto `coleccion`, y restando 1 porque como siempre, los *arrays* en Java están indexados desde 0 hasta el número de elementos menos 1. A continuación, vuelca los contenidos del objeto `coleccion` en la salida estándar.

- La sección de entrenamiento, que crea un clasificador de tipo `PART`, lo entrena sobre el conjunto de datos albergados en el objeto `coleccion` usando el método `buildClassifier()`, que es común a cualquier algoritmo de entrenamiento, y finalmente vuelca el modelo construido a la salida estándar.
- La sección de lectura de un nuevo ejemplo, que es creado como objeto de tipo `Instance` con cinco atributos vacíos, y se va asignando a cada uno de los cuatro primeros los siguientes valores leídos de la línea de órdenes usando el método `setValue()`, que acepta como primer argumento el número del atributo y como segundo atributo el valor del mismo. El ejemplar creado se muestra por la salida estándar.
- Finalmente, la sección de clasificación, donde se determina la clase del nuevo ejemplo usando el método `classifiy Instance()` del objeto clasificador de tipo `PART`, que es un método común también a todos los clasificadores. El resultado de este método es un número real almacenado en un tipo `double` porque algunos clasificadores devuelven aproximaciones numéricas. En este caso, el número nos da el índice de la clase dentro de un *array* que alberga todos los valores posibles (*none* – ninguno, *soft* – blandas, *hard* – duras), y se muestra por pantalla usando el método `value()` del objeto de tipo atributo que se obtiene consultando el objeto `coleccion` con el método `classAttribute()`, después de hacer una conversión (*cast*) a tipo entero (`int`) de la variable `indiceClase`.

En este ejemplo, cabe reseñar que los métodos esenciales de un clasificador son `buildClassifier()` (para entrenar o construir un modelo sobre una colección de datos de entrenamiento), y `classifyInstance()` (para clasificar un nuevo ejemplar de acuerdo con el modelo). También conviene subrayar que la lectura de los datos del nuevo ejemplar se puede realizar por medio de un bucle genérico, en cuyo caso el programa funcionaría sobre cualquier colección de datos, y no sólo en aquellas en las que haya cinco atributos, todos ellos de tipo nominal (como es el caso de `contact-lenses.arff`). Sin embargo, se complicaría notablemente el ejemplo, ya que sería preciso comprobar el tipo de cada atributo que se lee y hacer el análisis y la asignación de manera dependiente de este tipo por medio de una estructura `switch-case`.

El programa se compila a código intermedio (*bytecode* de Java) por medio de la siguiente instrucción, que funciona sólo si hemos agregado el directorio donde está el archivo `weka.jar` a la variable de entorno `CLASSPATH`, para que Java pueda encontrar las dependencias:

```
javac EjemploWeka.java
```

Para ejecutar el programa, copiamos el archivo contact-lenses.arff al directorio donde estemos compilando el programa, que ahora contiene el archivo EjemploWeka.class, y hacemos la llamada a la Máquina Virtual de Java (java) con los parámetros adecuados, que son el nombre del archivo con los datos, y los atributos del nuevo ejemplar a clasificar:

```
java EjemploWeka contact-lenses.arff
young myope no reduced
```

En nuestro ejemplo, hemos tomado los datos del primer ejemplar de entrenamiento, cuya clase real es *none* (ninguna lente), ya que la tasa lacrimal es reducida (*reduced*) y se aplica la primera regla del clasificador PART. El resultado de la ejecución del programa se muestra en el listado 5, donde se presenta la colección de datos leída, el clasificador PART entrenado, los datos del ejemplar leído y su clase según PART, que es correcta (coincide con la clase del ejemplar en la colección de entrenamiento). Se puede observar que el ejemplar que se muestra en el ejemplo tiene `?` como valor del último atributo, es decir, la clase es desconocida a priori.

Conclusiones y continuación

Visto este ejemplo, creemos que queda demostrado que, con algunos conocimientos de programación, y las bibliotecas adecuadas, es perfectamente posible escribir programas que posean características de Inteligencia Artificial, y en particular, de Aprendizaje Automático. Los ejemplos que se han presentado han servido para introducir los conceptos básicos de este área, así como para aplicarlos con la herramienta WEKA. El programa de ejemplo muestra como es de sencillo desarrollar un programa que hace uso de esta biblioteca para exhibir capacidades predictivas. Sin embargo, el problema que aborda el programa de ejemplo es muy simple y con un interés colateral. Por ello, en los próximos números de esta revista iremos desarrollando otros programas también sencillos que aplican los conceptos anteriores y la biblioteca WEKA para hacer cosas más interesantes, como un filtro bayesiano de correo basura, o un recomendador de *feeds* (que sirve para clasificar las *feeds* a las que uno está suscrito por su interés).



En la red

- What is Artificial Intelligence, John McCarthy
  http://www-formal.stanford.edu/jmc/whatisai/whatisai.html,
- WEKA
  http://www.cs.waikato.ac.nz/ml/weka/,
- Orange
  http://www.ailab.si/orange/,
- RapidMiner Community Edition
  http://www.rapidminer.com/,
- Proyecto R
  http://www.r-project.org/.



Sobre el autor

José María Gómez Hidalgo es Doctor en Ciencias Matemáticas, y director de I+D en la empresa de seguridad Optenet. Ha sido profesor e investigador de la Universidad Europea de Madrid, dirigiendo el Departamento de Sistemas Informáticos, e impartiendo asignaturas relacionadas con la seguridad informática, la Inteligencia Artificial, la programación y la algorítmica. Ha liderado varios proyectos de investigación centrados en herramientas de filtrado de contenidos Web y de filtrado de correo basura o *spam*. Dentro de la seguridad, su especialidad es la aplicación de técnicas de Inteligencia Artificial, y ha publicado múltiples artículos científicos, e impartido conferencias nacionales e internacionales sobre el correo basura y el filtrado Web. Su página web es *http://www.esp.uem.es/jmgomez*, y su blog es *http://jmgomezhidalgo.blogspot.com*.

Theremín Virtual: Un instrumento musical de nueva generación

Lino García Morales

El *Theremín* es según la Wikipedia uno de los primeros instrumentos musicales electrónicos. Los Theremines **originales se fabricaron con válvulas de vacío. Posteriormente, desde la aparición del transistor, multitud de firmas** comercializan versiones transistorizadas mucho más robustas, estables, adecuadas para el transporte, y de menor consumo eléctrico. Sin embargo la tecnología digital ha avanzado mucho y comienzan a aparecer diversos proyectos **de Theremines que reemplazan a sus antecesores por dispositivos fácilmente asequibles y que, en definitiva, portan** la implementación física, electrónica, a una implementación virtual, programada.



linux@software.com.pl

En este artículo se analiza la construcción de un Theremín mediante una webcam barata y un entorno de programación libre y gratuito como es Processing (*http://processing.org/*).

Introducción

El timbre original del Theremín era muy peculiar (algo entre un violonchelo y una voz humana) y aunque, en dependencia de la tecnología utilizada (MIDI, por ejemplo), es posible dotarle, virtualmente, de cualquier timbre (sintetizando su sonido desde un sampler o muestreador), es ese sonido original el más deseado quizá, por el uso que ha venido teniendo en la industria del cine (a menudo en películas de ciencia-ficción y terror). El Theremín fue un instrumento revolucionario en su época y llegó a conseguir la suficiente credibilidad como instrumento solista en un entorno orquestal e incluso en la música pop y rock (en la película *The Song Remains the Same* de Led Zeppelin, Jimmy Page improvisa efectos de sonido con un rústico Theremín de una antena conectado a efectos de retardo, durante la canción *Whole Lotta Love*).

Este aparato debe el nombre a su autor León Thérémin y consiste en una caja con dos antenas que produce un sonido relacionado con la posición de ambas manos respecto a cada una de las antenas. La antena izquierda es horizontal y sirve para controlar el volumen: cuanto más cerca de la misma esté la mano izquierda, más baja el volumen, y viceversa. La antena derecha suele ser recta y en vertical, y sirve para controlar la frecuencia: cuánto más cerca esté la mano derecha de la misma, más agudo será el sonido producido. El Theremín *se toca sin tocarlo*. Se basa en una técnica conocida como heterodino o frecuencia de batido para la que se necesitan dos osciladores que producen ondas cuyas frecuencias son superiores a las que el oído humano puede percibir. El primer oscilador produce una frecuencia fija de 170 kHz mientras el segundo una frecuencia que varía entre los 168 y 170 kHz. Estas ondas interfieren entre sí y dan lugar a nuevas ondas audibles, cuyo tono y volumen se controla moviendo las manos alrededor de las antenas del instrumento, pero sin hacer contacto con ella. La resta de las señales generadas por ambos osciladores, da como resultado una onda cuyo ran-



Figura 1. Lev Sergeyevitch Temev, 1896-1993, (León Thérémin) interpretando con su instrumento electrónico

go (0 Hz a 2 KHz) queda dentro del intervalo audible por el ser humano (20 Hz a 20 kHz). Esto hace que sea un instrumento muy difícil de tocar (tanto volumen como altura del sonido responden a cualquier movimiento del intérprete), por lo que su ejecución requiere una gran habilidad y maestría. En el enlace *http://axxon.com.ar/zap/223/c-Zapping0223.htm* puede encontrar más información acerca de este curioso instrumento.

Theremín Virtual

El Theremín virtual genera los valores de tono y volumen necesarios y, a partir de ellos, el sonido, en base a información que extrae de una imagen. La figura 2 muestra varios ejemplos de Theremines digitales basados en webcam. *Peripheral MIDI Controller* (*http://pmidic.sourceforge.net/*) genera información MIDI a partir de la detección de la posición de un objeto conocido (la luz que produce un LED, Light-Emitting Diode). *Digital Theremín* (*http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/ug2004/abs/u1sk.htm*) de Sel-Vin Kuik explora técnicas de seguimiento de la mano computacionalmente eficientes para



Figura 2. Etherwave® Theremín 110V fabricado por MOOG

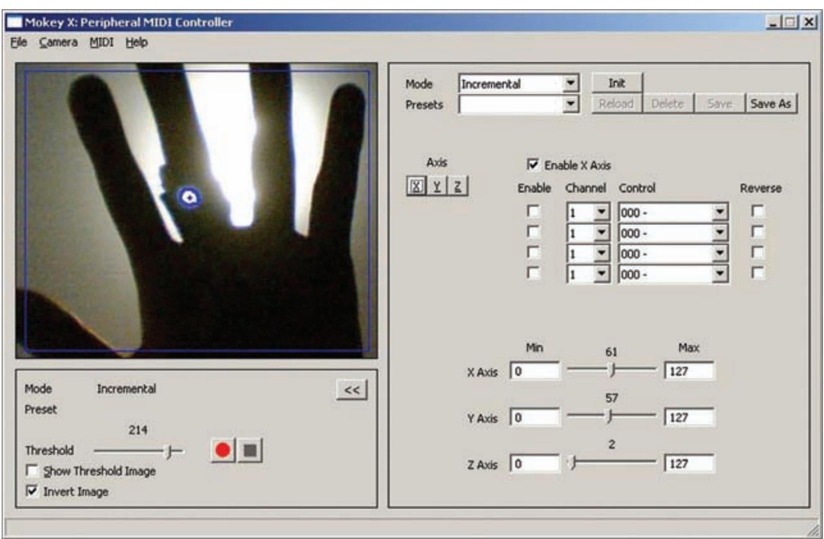


Figura 3. Peripheral MIDI Controller

correr en tiempo real. *Virtual Theremín* (*http://homepages.ihug.com.au/~squires/vt/*) de Edward Squires también identifica las manos y sigue su movimiento.

En general, cualquier sistema basado en el procesamiento de la imagen requiere de dos procesos independientes: *calibración* y *ejecución*. La calibración es un proceso muy importante. Su objetivo es, en general, establecer una relación conocida y adecuada entre el mundo real y el mundo virtual. En modelos sencillos se utiliza para extraer la escena principal o de fondo, sin el ejecutante, y para identificar los colores o patrones de piel mediante modelos estadísticos, etc. Sel-Vin Kuik, por ejemplo, utiliza el algoritmo CAMSHIFT (Continuously Adaptative Mean-Shift), para seguir zonas de la imagen con un determinado patrón de color. En modelos más complejos, como los de visión estereoscópica, sirve para obtener los parámetros intrínsecos (internos) de la cámara, tales como la distancia focal, los factores de distorsión y los puntos centrales del plano de la imagen.

El proceso de ejecución puede ser más o menos complejo. Su objetivo es la identificación de un patrón dentro de la imagen. Asignarle una coordenada espacial <*x*, *y*> en el caso de una sola cámara <*x*, *y*, *z*> para el caso de visión estereoscópica y transformar este dato a un espacio musical, como puede ser <*tono*, *volumen*> o <*panorámica*, *tono*, *volumen*>. Esta transformación normalmente corresponderá a un cambio de escala que ajuste la sensibilidad, tesitura y espacialidad del instrumento.

La ventaja de utilizar Processing (del que se ha publicado un artículo en el número anterior) es porque es de código abierto, fácil de instalar y utilizar y perfectamente soportado en Linux). Para la implementación del Theremín Virtual en Processing se utilizarán dos bibliotecas, igualmente de código abierto.

El código completo del Theremín en Processing se dividirá en varios listados para poder explicar mejor su funcionamiento. El Listado 1 muestra cuáles son las bibliotecas a incluir. La primera, BlobDetection (*http://v3ga.net/processing/BlobDetection/*), sirve para detectar áreas o regiones en la imagen más brillantes u oscuras que las que le rodean (llamadas *Blobs*). La segunda, Beads (*http://www.beadsproject.net/?page_id=9*), es necesaria para controlar y sintetizar el sonido.

El Listado 2 muestra las variables globales necesarias. Las tres primeras líneas definen variables para el control del sonido. a es la



Figura 4. Digital Theremin de Sel-Vin Kuik



Figura 5. Virtual Theremin de Edward Squires

```
import net.beadsproject.beads.core.*;
import net.beadsproject.beads.ugens.*;
import net.beadsproject.beads.data.*;
import net.beadsproject.beads.analysis.segmenters.*;
import net.beadsproject.beads.analysis.*;
import net.beadsproject.beads.events.*;
import net.beadsproject.beads.data.buffers.*;
import net.beadsproject.beads.analysis.featureextractors.*;
import net.beadsproject.beads.miscexperiments.*;
import processing.video.*;
import blobDetection.*;
```

Listado 1. Uso de las bibliotecas necesarias

```
AudioContext ac;
Glide carrierFreq, modFreqRatio, ge;
Gain g;
Capture a;
BlobDetection bd;
float xPos = 0, yPos = 0;
```

Listado 2. Variables globales

```
void setup() {
  size(640, 480, P2D);
  colorMode(HSB, 360, 100, 100);
  background(0);

  // List all available capture devices to the console
  println(Capture.list());

  // Specify your own device capture
  a = new Capture(this, width, height, 24);
  //a.settings();
  bd = new BlobDetection(width, height);
  bd.setPosDiscrimination(true);
  bd.setThreshold(0.2f);     // will detect bright areas whose luminosity >
0.2f;

  smooth();
  noCursor();
  loadPixels();

  ac = new AudioContext();
  //this time we use the Glide object because it smooths the mouse input.
  carrierFreq = new Glide(ac, 500);
  modFreqRatio = new Glide(ac, 1);
  ge = new Glide(ac, 1);
  Function modFreq = new Function(carrierFreq, modFreqRatio) {
    public float calculate() {
      return x[0] * x[1];
    }
  };
  WavePlayer freqModulator = new WavePlayer(ac, modFreq, new SineBuffer().
getDefault());
  Function carrierMod = new Function(freqModulator, carrierFreq) {
    public float calculate() {
      return x[0] * 400.0 + x[1];
    }
  };
  WavePlayer wp = new WavePlayer(ac, carrierMod, new SineBuffer().
getDefault());
  g = new Gain(ac, 1, ge);
  g.addInput(wp);
  ac.out.addInput(g);
}
```

Listado 3. Definición de la función setup()

```
void draw() {
  if(a.available()) {
    a.read();
    a.loadPixels();
    arraycopy(a.pixels, pixels);

    bd.computeBlobs(pixels);
    drawBlobsAndEdges(true, true);
    updatePixels();
  }

  carrierFreq.setValue((float)xPos/ width * 1000 + 50);
  modFreqRatio.setValue((1 - (float)yPos/ height) * 10 + 0.1);
  ge.setValue((1 - (float)yPos/ height)* 0.5);
}
```

Listado 4. Definición de la función draw(). El bloque draw() corre infinitamente

variable que se utilizará para importar el vídeo, en tiempo real, desde la webcam. xPos e yPos son variables auxiliares para seguir, de la manera más simple, el centroide del área mayor que se forme (Blob). El Listado 3 muestra el código de inicialización en la función setup().Esta función corre sólo una vez. Normalmente se utiliza para crear e inicializar todos los objetos necesarios, definir el modo de trabajo, tamaño de ventana, velocidad de captura, etc. Primero se define el tamaño de la ventana (640x480). La ventana, cuanto más grande, mayor consumo de tiempo de proceso. A continuación el modo de trabajo. En este caso se ha elegido el espacio de color HSB (Hue, Saturation, Brigthness) de la figura 3. Observe que se establece un rango de valores para el color de 360º, mientras que para la saturación y brillo un rango del 0 al 100%.

La saturación define la cantidad de blanco o negro en porcentaje (0% afuera, 100% dentro). Un color con mucho blanco es más pastel mientras que con mucho negro es opaco. El brillo define la intensidad del color en porcentaje (0% abajo, 100% arriba). Trabajar en este modo puede resultar mucho más cómodo que en RGB (Red, Green, Blue). A continuación se crea el objeto de captura (webcam). Observe que la frecuencia de adquisición es de 24 cuadros (*frames*) por segundo. El objeto que detecta los Blobs necesita un umbral de luminosidad para diferenciar las áreas. En este caso se suministra un valor bajo igual a 0.2 que detecte sólo zonas muy oscuras. Es destacable que estos parámetros se deben ajustar en dependencia de la iluminación del entorno donde se va a utilizar. No existe un conjunto que funcione bien en cualquier condición.

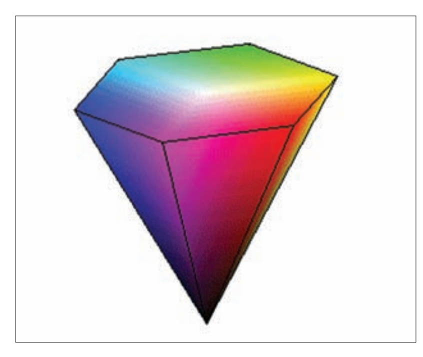


Figura 6. Cono del color del espacio HSB

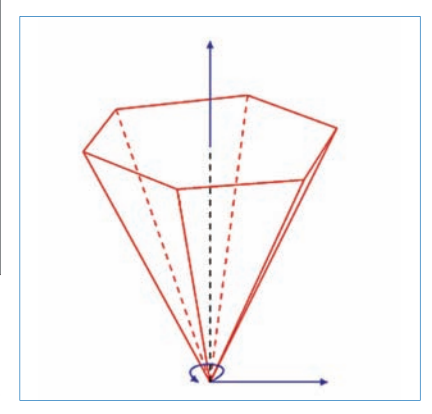


Figura 7. El valor del color o tonalidad (*hue*) varía al girar alrededor del *cono* (de 0 a 360º)

```
void keyPressed() {
  ac.stop();
  ac.start();
}
```

Listado 5. Función que captura cualquier acción desde el teclado

```
void drawBlobsAndEdges(boolean drawBlobs, boolean drawEdges) {
  xPos = 0;
  yPos = 0;

  noFill();
  Blob b;
  EdgeVertex eA,eB;
  for(int k = 0 ; k < bd.getBlobNb(); k++) {
    b = bd.getBlob(k);
    if(b != null) {
      // Edges
      if(drawEdges) {
        strokeWeight(3);
         stroke(0, 99, 99);
         for(int m = 0; m <b.getEdgeNb(); m++) {
          eA = b.getEdgeVertexA(m);
           eB = b.getEdgeVertexB(m);
           if(eA != null && eB != null)
            line(eA.x* width, eA.y* height, eB.x* width, eB.y* height);
        }
      }
      // Blobs
      if (drawBlobs) {
        strokeWeight(1);
         stroke(100, 99, 99);
        rect(b.xMin* width, b.yMin* height, b.w* width, b.h* height);
        if(b.x* width* b.y* height > xPos* yPos) {
          xPos = b.x* width;
          yPos = b.y* height;
        }
      }
    }
  }
}
```

Listado 6. Función que representa gráficamente los Blobs sobre la imagen capturada

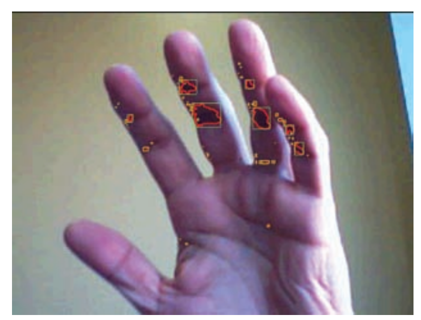


Figura 8. Detección de Blobs en una trama de ejemplo: Mano derecha del autor

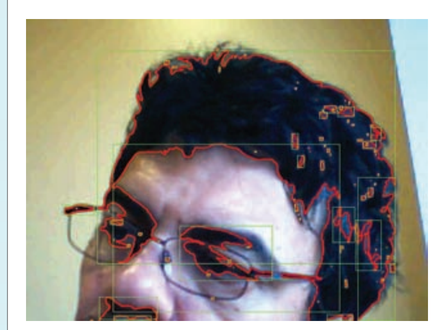


Figura 9. Detección de Blobs en una trama de ejemplo: Cabeza del autor

El objeto de audio, que provee la biblioteca del proyecto *beads*, se ha configurado para modular en frecuencia una portadora de 500 Hz. Para ello se construyen tres controles: frecuencia de la portadora (carrierFreq), índice de modulación (modFreqRatio) y ganancia (ge). La frecuencia aumentará según el objeto seguido se desplace a la derecha y disminuirá hacia la izquierda. El volumen (controlado por la ganancia) y el índice de modulación aumentará hacia arriba y disminuirá hacia abajo.

El Listado 4 define el comportamiento continuo del programa. Carga la imagen de la cámara, la copia al fondo (*background*), estima los *Blobs*, y los representa sobre la imagen mediante la función dranBlobsAndEges(true, true); de la biblioteca BlobCapture. Esta función ha sido manipulada por el autor para obtener los valores de xPos e yPos. Observe cómo se relacionan los tres controles creados para controlar la síntesis de audio con la posición del centroide del Blob de mayor área.

El audio no sonará hasta que no se pulse cualquier tecla como muestra el Listado 5.

La Figura 4 muestra una trama durante la reproducción. Observe que el programa detecta las zonas más oscuras (por debajo de 0.2 de iluminación en Listado 3) y, de todos los Blobs, se selecciona el de mayor área. Los Blobs son representados según el código del Listado 6.

Conclusiones

El Theremín virtual es sólo un ejemplo de aplicación que procesa vídeo para generar sonido realizado con pocas modificaciones de distintos ejemplos de las bibliotecas. La modulación que se utiliza aquí casualmente recrea un sonido *parecido* al del Theremín original pero esta sencilla aplicación es susceptible de modificar para conseguir un seguimiento más preciso, la escala adecuada de frecuencias e incluso el timbre. En lugar de este simple modo de síntesis de sonido se pueden generar eventos MIDI (Musical Instrument Digital Interface) que controlen máquinas más complejas con cualquier otro tipo de sonido. El seguimiento del Blob de mayor área es sólo una forma muy sencilla de implementación para ilustrar el comportamiento del Theremín. Modelos más complejos pueden seguir patrones como la piel, la punta de los dedos, o incluso, con reconocimiento facial, parte de la cara. La intercalación de procesos como: conversión monocromática de la imagen, substracción del fondo, aplicación de filtros de mediana y/o filtros morfológicos de dilatación y erosión para la eliminación de ruido, etc. pueden ayudar a mejorar el seguimiento de determinado patrón. Otro aspecto mejorable podría ser el propio seguimiento de un patrón, mediante técnicas de visión estereoscópica, en tres dimensiones.

Conecte una webcam barata, cree un nuevo Sketch, coja todos los códigos de los listados y júntelos en uno solo, dele al botón y tendrá un Theremín Virtual para empezar a jugar y cuestionarse qué podría hacer para llegar más lejos mientras aprende.



Sobre el autor

Lino García Morales es Graduado en Ingeniería en Control Automático, Máster en Sistemas y Redes de Comunicaciones y Doctor por la Universidad Politécnica de Madrid. Ha sido profesor en el Instituto Superior de Arte de la Habana, la Universidad Pontificia "Comillas" y la Universidad Menéndez Pelayo.

Actualmente es profesor de la Escuela Superior de Arte y Arquitectura y de la Escuela Superior Politécnica de la Universidad Europea de Madrid y Director del Máster Universitario en Acústica Arquitectónica y Medioambiental. Músico, escritor y científico, lidera un grupo de investigación transdisciplinar en la intersección Arte, Ciencia y Tecnología. Ha disfrutado de Becas por la Agencia Española de Cooperación Internacional, FUNDESCO, el Consejo Superior de Investigaciones Científicas (CSIC) y la Universidad Politécnica de Madrid.

Herramientas forenses para la adquisición de datos

Alonso Eduardo Caballero Quezada

Las principales etapas en una metodología de cómputo forense implican cuatro fases: recolección, preservación, análisis y presentación. La etapa de recolección es donde los objetos que se consideran de valor como evidencia son identificados y recolectados. Estos objetos son datos digitales en forma de unidades de disco, unidades de memorias flash, u otras formas de medios digitales y datos. En el presente artículo se exponen casi todas las herramientas que pueden ser utilizadas en GNU/Linux, para obtener o adquirir datos que pueden contener evidencia digital.



linux@software.com.pl

En el presente artículo se realizará una evaluación en el funcionamiento y desempeño de la mayoría de herramientas disponibles en GNU/Linux para realizar la copia exacta de evidencia digital. El escenario de utilización de las herramientas es el siguiente: todas las herramientas serán configuradas, compiladas y utilizadas con sus definiciones por defecto, excepto que sea necesario alguna compilación o parámetro especial para su funcionamiento. Cada herramienta funcionará tratando de recuperar información desde un disco flexible de 3 1/2 que data del año 2005 y que su conservación no ha sido realizada en las más óptimas condiciones.

Disco Flexible (Floppy Disk)

Tal vez los discos flexibles para una gran cantidad de lectores les sea poco familiar dada la masiva utilización de dispositivos con grandes capacidades de almacenamiento, llámese unidades USB o *USB memory sticks*. Pero aun estos discos flexibles se siguen comercializando, y por allí conservo algunos, entre ellos uno, el cual será objeto de análisis para evaluar las herramientas de adquisición y réplica del presente artículo. Se obvia la historia y evolución de los discos flexibles y a continuación se detallan sus principales características técnicas.

Está hecho de una pieza de plástico cubierta con un material magnético en ambos lados, tal como la muestra la Figura 1. Su forma es la de un disco. Las pistas están alineadas en anillos concéntricos, así el software puede saltar desde el "archivo 1" al "archivo 19" sin tener que desplazarse por los archivos 2 al 18. El disco gira y las cabezas se mueven hacia las pistas correctas, proporcionando lo que se conoce como acceso directo al almacenamiento.

Los datos son colocados en círculos magnéticos concéntricos llamados pistas. Un disco de 3.5 pulgadas tiene 80 pistas por lado. Los discos de doble lado tienen datos en ambos lados. Por lo tanto hay dos lados físicos. Una cabeza está dedicada a un lado. Las cabezas se mueven hacia adentro y afuera sobre las pistas para leer/escribir datos mientras el disco gira. Ambos lados forman un Cilindro, es decir la Pista 0, Lado 0 + Pista 0, Lado 1 = Cilindro 0.

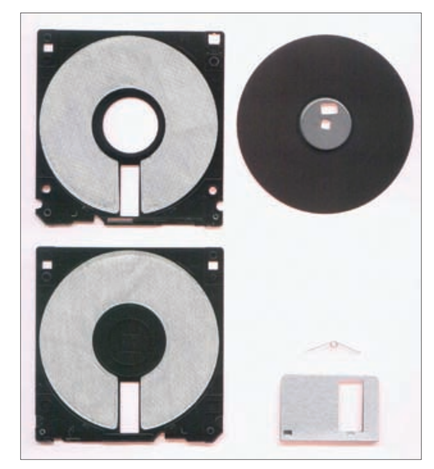


Figura 1. Imagen del interior de un disco flexible de 3 1/2

Cada pista se divide en unidades de almacenamiento llamados sectores. El sistema de archivos FAT utiliza 512 sectores de manera exclusiva. El número de sectores por pista varía dependiendo del medio y del formato. La numeración de los sectores físicos inicia con el número 1 en el inicio de cada pista y lado. Los sectores son la unidad más pequeña que puede ser direccionada, es decir, el espacio más pequeño en el cual puede ser realizada una escritura y debe tener una dirección antes de que el sector sea leído. Y finalmente un cluster es un grupo de sectores.

Entonces resumiendo, un disco flexible en números:

```
Número de sectores = 1 - XX (1- 18).
18 en total.
Pistas / Cilindros = 0 - XX (0 -
79). 80 en total.
Cabezas / Lados = 0 - XX  (0 - 1). 2
en total.
```

Capacidad de almacenamiento = 80 x 2 x 18 x 512, es decir el equivalente a 1474560 bytes. Dado que 1440 KiB (1 KiB = 1024 bytes) o 1.44 MB.

dd

En cómputo forense es de mucha utilidad cuando se necesita que un patrón magnético de un disco completo sea preservado como una copia exacta de bytes. Utilizando el comando *cp* esto no es posible debido a que los datos de archivos borrados siguen estando presentes físicamente en el disco y no son visibles mediante la interfaz del sistema de archivos.

Dd, o denominado algunas veces como GNU dd, es la herramienta más antigua de réplica que se ha utilizado. Aunque es totalmente funcional y requiere sólo de recursos mínimos para su ejecución, tiene carencias de algunas características útiles que se encuentran en la mayoría de replicadores modernos, tales como la obtención de metadatos, corrección de errores, *hashing* de porciones, y una interfaz amigable. dd es un programa en línea de comando que utiliza muchos argumentos oscuros en su línea de comando para controlar el proceso de réplica. Debido a que algunas de estas opciones son similares y, si se confunden, pueden destruir el medio de origen que el examinador está intentando duplicar, los usuarios deben ser cautelosos con la ejecución de este programa. El programa genera archivos de imagen en bruto que pueden ser leídos por la mayoría de programas.

Existen algunas bifurcaciones de dd para propósitos forenses incluyendo dcfldd, sdd, dd_rescue, ddrescue, rdd y una versión para Microsoft Windows que soporta la lectura de la memoria física.

Instalación de dd

dd es parte del paquete de *GNU Core Utilities* que son archivos básicos, shell y utilidades de manipulación de texto del sistema operativo GNU. Son las utilidades del núcleo, las cuales se espera que existan en cualquier sistema operativo. Por lo antes mencionado, dd se encuentra en cualquier distribución GNU/Linux. Si se desea realizar su creación desde las fuentes, se debe descargar el archivo *coreutils-7.6.tar.gz* y proceder a su configuración e instalación con los comandos: *./configure; make; make install*. En mi caso solamente he creado los binarios, pero no los he instalado. La Figura 2 muestra algunos de los binarios, entre ellos dd, que se han creado en el directorio donde se descomprimió el paquete *GNU Core Utilities*.

Utilización de dd

Se debe recordar que dd copia un archivo (desde una entrada estándar, hacia una salida estándar por defecto) con un tamaño de bloque de E/S que puede ser modificado, mientras de manera opcional realiza conversiones sobre éste. Entre los muchos operadores que acepta dd, se detallan los más usuales en cómputo forense.

La forma de utilización de dd es la siguiente: *./dd [operando] o dd [opción]:*

- if = *archivo* - Lee desde *archivo* en lugar de la entrada estándar.
- of = *archivo* - Escribe en un *archivo* en lugar de la salida estándar. A menos que *conv = notrunc* sea definido, dd trunca *archivo* a cero bytes (o al tamaño especificado con *seek =*).
- ibs = *bytes* - Define el tamaño de bloque de entrada a *bytes*. Esto hace que dd lea *bytes* por bloque. Por defecto es de 512 bytes.
- obs = *bytes* - Define el tamaño de bloque de salida a *bytes*. Esto hace que dd escriba *bytes* por bloque. Por defecto es de 512 bytes.
- bs = *bytes* - Define el tamaño de bloque de entrada y salida a *bytes*. Esto hace que dd lea y escriba *bytes* por bloque siendo imperativo ante cualquier definición de *ibs* u *obs*. Además si no se especifica la opción de transformación de datos *conv*, cada bloque de entrada es copiado hacia la salida como un bloque único sin añadir lecturas cortas.
- skip = *bloques* - Salta *bloques* ibs-bytes en el archivo de entrada antes de copiar.
- seek = *bloques* - Busca *bloques* obs-bytes en el archivo de salida antes de copiar.

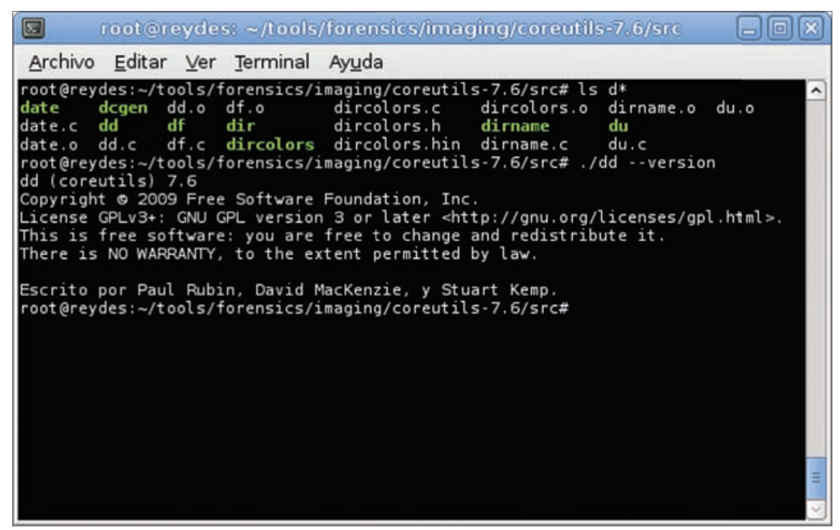


Figura 2. Listado del directorio donde se compiló el paquete GNU Core Utilities

- count = bloques - Copia *bloques* ibs-bytes desde el archivo de entrada, en lugar de todo hasta el final del archivo.
- conv = conversión[,conversión]... - Convierte un archivo especificado por el argumento(s) conversión. (Sin espacios alrededor de las comas).

A continuación se detallan algunos de los tipos de conversiones disponibles:

- notrunc = No trunca el archivo de salida.
- noerror = Continua después de leer errores.
- sync = Rellena cada bloque de entrada al tamaño de *ibs* con rastros de bytes de ceros. Cuando se usa con *block* o *unblock*, se rellena con espacios en lugar de bytes de ceros.
- fdatasync = Sincroniza los datos de salida sólo antes de finalizar. Esto fuerza a la escritura física de los datos de salida.
- sync = Utiliza E/S sincronizada para datos y metadatos.
- noatime = No actualiza el tiempo de acceso al archivo. Algunos sistemas de archivos antiguos de manera silenciosa ignoran esta opción, así es que es una buena idea probarlo en los archivos antes de confiar en esto.

Las cadenas de valor numérico (bytes y bloques) pueden ser seguidas por un multiplicador: b = 512, c = 1, w = 2, xm = m o cualquiera de los sufijos estándar de tamaño de bloque como k =1024.

Resultados de dd

El comando utilizado para realizar la copia bit a bit desde el disco flexible es:

```
./dd bs=512 conv=noerror,sync if=/
dev/fd0 of=tools/forensics/imaging/
floppy_dd.dd
```

dd copió 1474560 bytes del disco flexible en 576,9 segundos. Su hash SHA1 es el siguiente:

```
b153a22672f6d4788a2a7f5c28ff18264
a02f7f5  tools/forensics/imaging/
floppy_dd.dd
```

dcfldd

dcfldd es una versión mejorada de GNU dd desarrollada por el laboratorio de cómputo forense del Departamento de Defensa de los Estados Unidos. dcfldd fue creada originalmente por Nicholas Harbour. El programa solamente produce archivos imagen en bruto. Tiene algunas características muy útiles para los investigadores forenses tales como:

- *Hashing* al vuelo de los datos transmitidos. dcfldd puede hacer un hash de los datos de entrada tal y como estos están siendo transmitidos, ayudando a asegurar la integridad de los datos.
- Limpieza (*Wiping*) flexible de discos. dcfldd puede ser utilizado para limpiar discos rápidamente y si se desea con un patrón conocido.
- Verificación de la imagen/limpieza. dcfldd puede verificar que la unidad de origen es una correspondencia bit a bit del archivo de entrada especificado o patrón.
- Varias salidas. dcfldd puede generar salidas a varios archivos o discos al mismo tiempo.
- Salida dividida. dcfldd puede dividir la salida en varios archivos con mayor poder de configuración que el comando *split*.
- Salidas por tubería y registros. dcfldd puede enviar todos sus datos de registros y salidas a comando, así como archivos de manera nativa.

```
root@reydes: ~/tools/forensics/imaging/dcfldd-1.3.4-1
Archivo  Editar  Ver  Terminal  Ayuda
ChangeLog       full-write.h   Makefile       sha1.h        verify.c
config.guess    full-write.o   Makefile.am    sha1.o        verify.h
config.h        getpagesize.h  Makefile.in    sha2.c        verify.o
config.h.in     hash.c         md5.c          sha2.h        version-etc.c
config.log      hashformat.c   md5.h          sha2.o        version-etc.h
config.status   hashformat.h   md5.o          sizeprobe.c   version-etc.o
config.sub      hashformat.o   missing        sizeprobe.h   xalloc.h
configure       hash.h         NEWS           sizeprobe.o   xstrtol.c
configure.in    hash.o         output.c       split.c       xstrtol.h
copy.c          human.c        output.h       split.h       xstrtol.o
copy.h          human.h        output.o       split.o       xstrtoul.c
COPYING         human.o        pathmax.h      stamp-h1      xstrtoul.o
copy.o          INSTALL        pattern.c      sys2.h        xstrtoumax.c
dcfldd          install-sh     pattern.h      system.h      xstrtoumax.o
dcfldd.1        log.c          pattern.o      translate.c
root@reydes:~/tools/forensics/imaging/dcfldd-1.3.4-1# ./dcfldd --version
dcfldd (dcfldd) 1.3.4-1
Written by: dcfldd by Nicholas Harbour, GNU dd by Paul Rubin, David MacKenzie an
d Stuart Kemp.

Copyright (C) 1985-2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
root@reydes:~/tools/forensics/imaging/dcfldd-1.3.4-1#
```

Figura 3. Listado del directorio donde se ha compilado el paquete dcfldd-1.3.4-1.tar.gz

```
root@reydes: ~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-lin
Archivo  Editar  Ver  Terminal  Ayuda
-rw-r--r-- 1   12 300   808 2000-11-19 13:01 Makefile_de.man
-rw-r--r-- 1   12 300   678 2000-11-19 13:01 Makefile.man
drwxrwsr-x 3 root 300  4096 2009-10-14 20:34 OBJ
-r--r--r-- 1   12 300  8487 2001-07-20 13:09 sdd.1
-r--r--r-- 1   12 300 39533 2004-07-27 05:49 sdd.c
-r--r--r-- 1   12 300  9883 2001-07-20 13:09 sdd_de.1
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd# cd OBJ/
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ# ls
i686-linux-cc
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ# cd i686-linux-cc/
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-linux-cc# ls -l
total 96
-rw-r--r-- 1 root 300     1 2009-10-14 20:34 Dnull
-rwxr-xr-x 1 root 300 60637 2009-10-14 20:34 sdd
-rw-r--r-- 1 root 300  2317 2009-10-14 20:34 sdd.d
-rw-r--r-- 1 root 300 24228 2009-10-14 20:34 sdd.o
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-linux-cc# ./sdd --ve
rsion
sdd 1.52 (i686-pc-linux-gnu)

Copyright (C) 1984-2004 J�rg Schilling
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-linux-cc#
```

Figura 4. Listado del directorio donde se ubica el binario sdd después de su compilación

Instalación de dcfldd

ddcfldd puede ser obtenido desde los repositorios oficiales, en este caso para GNU/Linux Debian o Ubuntu, con los siguientes comandos: *apt-cache search dcfldd; apt-get install dcfldd*.

Si se desea realizar la instalación desde las fuentes, se debe descargar el archivo *dcfldd-1.3.4-1.tar.gz*, que corresponde a su más reciente versión estable. Para proceder a su configuración e instalación se utilizan los siguientes comandos: *./configure, make; make install*. Nuevamente sólo se han creado los binarios, mas no he procedido a su instalación. La Figura 3 muestra el listado del directorio donde se ha llevado a cabo el procedimiento anteriormente descrito.

Utilización de dcfldd

dcfldd es una versión mejorada de dd, y como tal conserva muchas de sus opciones. A continuación se detallan algunas de las opciones más frecuentes utilizadas en cómputo forense.

La forma de utilización de dcfldd es la siguiente: *./dcfldd [opción]:*

```
bs = bytes - Fuerza ibs = bytes y
obs = bytes
cbs = bytes - Convierte bytes bytes
a la vez.
```

Conv = *palabrasclave* - Convierte el archivo tal y como lo indica la lista de palabras clave separadas por comas. Las *palabrasclave* son similares a las definidas para las conversiones de dd. Por ejemplo; *noerror*, continua después de los errores de lectura; *sync*, rellena cada bloque de entrada con NULs al tamaño *ibs*, cuando es utilizado con *block* o *unblock* rellena con espacios en lugar de NULLs; *notrunc*, no trunca el archivo de salida.

- count = *bloques* - Copia solamente *bloques* bloques de entrada.
- ibs = *bytes* - Lee *bytes* bytes a la vez.
- if = *archivo* - Lee desde el *archivo* en lugar de stdin
- obs = bytes - Escribe *bytes* bytes a la vez
- of = *archivo* - Escribe en *archivo* en lugar de la salida estándar. *of* = *archivo* puede ser utilizado varias veces para escribir la salida a múltiples archivos de manera simultánea.
- of := *comando* - Ejecuta y escribe la salida para que sea procesado por *comando*.
- seek = *bloques* - Salta *bloques* bloques de tamaño-obs al inicio de la salida.
- skip = *bloques* - Salta *bloques* bloques de tamaño-ibs al inicio de la entrada.
- errlog = *archivo* - Envía mensajes de error a *archivo* en lugar de stderr.
- hashwindow = *bytes* - Realiza un hash cada *bytes* cantidad de datos.
- hash = nombre - Ya sea, MD5, SHA1, SHA256, SHA384, o SHA512. El algoritmo por defecto es MD5. Para seleccionar que varios algoritmos se ejecuten de manera simultánea se debe ingresar los nombres en una lista separada por comas.
- hashlog = *archivo* - Envía la salida del hash MD5 a *archivo* en lugar de stderr, si se están utilizando varios algoritmos hash se debe de enviar cada uno a un archivo separado utilizando la convención *Algoritmolog* = *archivo*, por ejemplo *md5log* = *archivo1*, *sha1log* = *archivo2*, etc.
- split = *bytes* – Escribe la cantidad cada *bytes* de datos a un nuevo archivo. Esta operación se aplica a cualquier *of* = *archivo* que sigue.
- vf = *archivo* - Verifica que *archivo* corresponda con la entrada especificada.
- verifylog = archivo - Envía los resultados de la verificación a *archivo* en lugar de stderr.

```
root@reydes: ~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-lin
Archivo  Editar  Ver  Terminal  Ayuda
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2736 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2737 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2738 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2739 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2740 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2741 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2742 not read correctly.
./sdd: Input/output error. Error reading '/dev/fd0'.
./sdd: Block 2743 not read correctly.
./sdd: 264 Block(s) not read correctly.
./sdd: Read  2880 records + 0 bytes (total of 1474560 bytes = 1440.00k).
./sdd: Wrote 2880 records + 0 bytes (total of 1474560 bytes = 1440.00k).
[illegible]
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-linux-cc# sha1sum ..
/../../../floppy_sdd.dd
7ea25506e5d5c8e116be227aa59b06166aefda97  ../../../../floppy_sdd.dd
root@reydes:~/tools/forensics/imaging/sdd-1.52/sdd/OBJ/i686-linux-cc#
```

Figura 5. Culminación del proceso de la copia bit a bit utilizando sdd

```
root@reydes: ~/tools/forensics/imaging/dd_rescue
Archivo  Editar  Ver  Terminal  Ayuda
root@reydes:~/tools/forensics/imaging/dd_rescue# ls -l
total 60
-rw-r--r-- 1 548 users 18007 2000-01-14 12:02 COPYING
-rw-r--r-- 1 548 users 26989 2007-08-26 08:42 dd_rescue.c
-rw-r--r-- 1 548 users  1044 2007-08-26 08:36 Makefile
-rw-r--r-- 1 548 users  5578 2006-07-23 06:50 README.dd_rescue
root@reydes:~/tools/forensics/imaging/dd_rescue# make
gcc -O2 -Wall -g  -DVERSION=\"1.14\" dd_rescue.c -o dd_rescue
dd_rescue.c: En la función 'cleanup':
dd_rescue.c:264: aviso: se descarta el valor de devolución de 'pwrite', se decla
ró con el atributo warn_unused_result
dd_rescue.c: En la función 'copyfile':
dd_rescue.c:375: aviso: se descarta el valor de devolución de 'pwrite', se decla
ró con el atributo warn_unused_result
root@reydes:~/tools/forensics/imaging/dd_rescue# ls
COPYING  dd_rescue  dd_rescue.c  Makefile  README.dd_rescue
root@reydes:~/tools/forensics/imaging/dd_rescue# ls -l
total 120
-rw-r--r-- 1  548 users 18007 2000-01-14 12:02 COPYING
-rwxr-xr-x 1 root root  56794 2009-10-14 21:32 dd_rescue
-rw-r--r-- 1  548 users 26989 2007-08-26 08:42 dd_rescue.c
-rw-r--r-- 1  548 users  1044 2007-08-26 08:36 Makefile
-rw-r--r-- 1  548 users  5578 2006-07-23 06:50 README.dd_rescue
root@reydes:~/tools/forensics/imaging/dd_rescue# ./dd_rescue --version
```

Figura 6. Fin de la compilación de dd_rescue y listado del directorio

```
root@reydes: ~/tools/forensics/imaging/dd_rescue
Archivo  Editar  Ver  Terminal  Ayuda
             +curr.rate:        0kB/s, avg.rate:        2kB/s, avg.load:  0.0%
dd_rescue: (warning): /dev/fd0 (1370.5k): Input/output error!

dd_rescue: (info): ipos:     1371.0k, opos:     1371.0k, xferd:      1371.0k
                  * errs:    254, errxfer:     127.0k, succxfer:      1244.0k
             +curr.rate:        0kB/s, avg.rate:        2kB/s, avg.load:  0.0%
dd_rescue: (warning): /dev/fd0 (1371.0k): Input/output error!

dd_rescue: (info): ipos:     1371.5k, opos:     1371.5k, xferd:      1371.5k
                  * errs:    255, errxfer:     127.5k, succxfer:      1244.0k
             +curr.rate:        0kB/s, avg.rate:        2kB/s, avg.load:  0.0%
dd_rescue: (warning): /dev/fd0 (1371.5k): Input/output error!

dd_rescue: (info): ipos:     1440.0k, opos:     1440.0k, xferd:      1440.0k
                  * errs:    256, errxfer:     128.0k, succxfer:      1312.0k
             +curr.rate:       22kB/s, avg.rate:        2kB/s, avg.load:  0.0%
Summary for /dev/fd0 -> /media/hda3/tools/forensics/imaging/floppy_dd_rescue.dd:
dd_rescue: (info): ipos:     1440.0k, opos:     1440.0k, xferd:      1440.0k
                    errs:    256, errxfer:     128.0k, succxfer:      1312.0k
             +curr.rate:       22kB/s, avg.rate:        2kB/s, avg.load:  0.0%
[illegible]
root@reydes:~/tools/forensics/imaging/dd_rescue# sha1sum ../floppy_dd_rescue.dd
ca52119c24df55a7b4556b21052aa2a1387e987d  ../floppy_dd_rescue.dd
root@reydes:~/tools/forensics/imaging/dd_rescue#
```

Figura 7. Culminación del proceso de la copia bit a bit utilizando dd_rescue

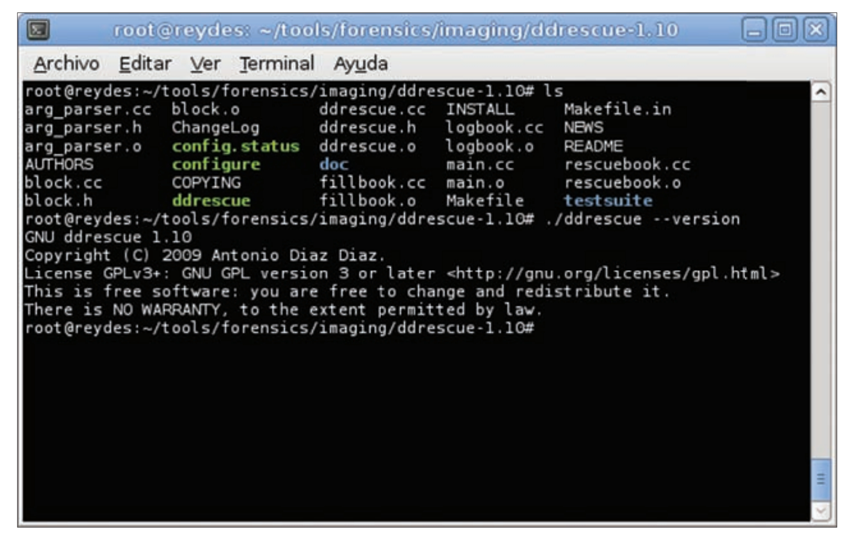


Figura 8. Listado del directorio donde se ha compilado GNU ddrescue

Resultados de dfcldd

El comando utilizado para realizar la copia bit a bit desde el disco flexible es:

```
./dcfldd conv=noerror,sync
errlog=floppydcfldd.err bs=512
hash=sha1 hashlog=floppydcfldd.sha1
if=/dev/fd0 of=tools/forensics/
imaging/floppy_dcfldd.dd
```

dfcldd copió 1474560 bytes del disco flexible en 1023,34 segundos. Su hash SHA1 es el siguiente, el cual se ha obtenido del archivo *floppydcfldd.sha1*:

```
64c86d62ae984548683ac0ebb6611a20f428
0f68  tools/forensics/imaging/floppy_
dcfldd.dd
```

El archivo *floppydcfldd.err* contiene el registro de los errores generados durante el proceso. La parte final de este archivo se presenta a continuación:

```
./dcfldd:/dev/fd0: Input/output error
2472+270 records in
2742+0 records out
./dcfldd:/dev/fd0: Input/output error
2472+271 records in
2743+0 records out
2608+272 records in
2880+0 records out
```

sdd

sdd es un reemplazo para el programa llamado *dd*. sdd es mucho más rápido que dd en casos donde el tamaño del bloque de entrada (ibs) no es igual al tamaño del bloque de salida (obs). Las estadísticas se entienden mejor que las de *dd*. Medida de tiempo disponible, la opción *-time* imprime la medida de tiempo de la velocidad de transferencia y estadísticas disponibles en cualquier momento sin SIGOUT. Soporte para el protocolo RMT (Servidor de Cinta Remoto) lo cual hace Entrada/Salida remota rápida y fácil.

Instalación de sdd

sdd puede ser obtenido desde los repositorios oficiales, en este caso para GNU/Linux Debian o Ubuntu, con los siguientes comandos: *apt-cache search sdd; apt-get install sdd.*

Para realizar la instalación desde las fuentes, se procede a descargar el archivo *sdd-1.52.tar.gz* que corresponde a la más reciente versión de sdd. Sugiero la lectura de dos archivos, *README.linux* e *INSTALL*, para proceder a la configuración e instalación de sdd. En resumen, estos dos archivos detallan la siguiente información.

Instalar */usr/bin/Gmake* con el comando: *cp Gmake.linux /usr/bin/Gmake.*

Luego compilar el sistema llamando: */usr/bin/Gmake* o *./Gmake.linux..*

Luego de finalizada la compilación el binario de sdd estará ubicado en *sdd-1.52/sdd/OBJ/i686-linux-cc/.*

La Figura 4 muestra el listado del directorio donde se ubica el binario de sdd, además de mostrar su versión.

Utilización de sdd

El modo de utilización de sdd es: *./sdd opción=valor -bandera*

- if = *nombre* - Lee la entrada desde *nombre* en lugar de stdin.
- of = *nombre* - Escribe la salida hacia *nombre* en lugar de stdout.
- -inull - No lee entrada desde archivo (usa caracteres null).
- -onull - No escribe en la salida a cualquier archivo.
- ibs=#, obs=#, bs=# - Define el tamaño del *buffer* de entrada/salida o ambos a #.
- cbs=# - Define el tamaño del *buffer* de conversión a #.
- ivsize=#, ovsize=# - Define el tamaño de volumen de entrada/salida a #.
- count=# - Transfiere la mayoría de # registros de entrada.
- iseek=#, iskip - Busca/salta # bytes en la entrada antes de iniciar.
- oseek=#, oskip - Busca/salta # bytes en la salida antes de iniciar.
- seek=#, skip=# - Busca/salta # bytes en la entrada/salida entes de iniciar.
- ivseek=#, ovseek=# - Busca # bytes en el volumen de entrada/salida antes de iniciar.
- -notrunc - No trunca el archivo de salida existente.
- -pg - Imprime un punto en cada escritura para indicar progreso.
- -noerror - No se detiene en un error.
- -noerrwrite - No escribe los sectores que no son leídos correctamente.
- -noseek - No busca.
- try=# - Define la cuenta de intentos de error a # si -noerror (por defecto 2).
- -debug - Imprime mensajes de depuración.
- -fill - Llena cada registro con ceros hasta obs.

Tabla 1. Los resultados obtenidos al aplicar todas las herramientas incluidas para obtener una imagen bit a bit del floppy disk

Programa	Bytes Copiados	Tiempo (Seg)	Hash (SHA1)
dd	1474560	577	b153a22672f6d4788a2a7f5c28ff18264a02f7f5
dfcldd	1474560	1023	64c86d62ae984548683ac0ebb6611a20f4280f68
sdd	1474560	1936	7ea25506e5d5c8e116be227aa59b06166aefda97
dd_rescue	1474560	783	ca52119c24df55a7b4556b21052aa2a1387e987d
ddrescue	1474560	559	ce6027744b370babf50f3e9300b3fa0e4b6178e4
rdd	1474560	155	c7b43564d85528d95e8396512933e30124e6c131

- -swab, -block, -unblock, -lcase, -ucase, -ascii, -ebcdic, -ibm.

Resultados de sdd

El comando utilizado para realizar la copia bit a bit desde el disco flexible es:

```
./sdd bs=512 if=/dev/fd0 of=tools/
forensics/imaging/floppy_sdd.dd -
noerror -pg -debug
```

sdd copió 1474560 bytes del disco flexible en 1936 segundos. Su hash SHA1 es el siguiente:

```
7ea25506e5d5c8e116be227aa59b06166
aefda97  tools/forensics/imaging/
floppy_sdd.dd
```

La Figura 5 muestra la parte final del proceso de copia. Como se puede apreciar en los mensajes, se muestran algunos errores de Entrada/Salida, es decir error al leer el dispositivo /dev/fd0. En total sdd no pudo leer correctamente 264 bloques.

dd_rescue

Como dd, dd_rescue hace una copia de datos desde un archivo o dispositivo de bloques a otro. Se pueden especificar posiciones en el archivo (llamado *seek* y *skip* en dd). A continuación algunas diferencias:

- dd_rescue no proporciona conversiones de caracteres.
- La sintaxis de los comandos es diferente. Llamar *dd_rescue -h*.
- dd_rescue no aborta cuando se presentan errores en el archivo de entrada, a menos que se especifique un número máximo de errores. Por lo tanto dd_rescue abortará cuando se alcance este número.
- dd_rescue no trunca el archivo de salida, a menos que se le solicite.
- Se puede indicar a dd_rescue a iniciar desde el final de archivo y moverse hacia atrás.
- Utiliza dos tamaños de bloque, un tamaño de bloque grande (suave) y un tamaño de bloque pequeño (duro). En caso de errores el tamaño cae de nuevo al pequeño y es promovido nuevamente después de un momento sin errores.

Propósito de dd_rescue

Las últimas tres características lo hacen adecuado para rescatar datos de un medio con errores, es decir, un disco duro con algunos sectores malos. ¿Por qué?

- Imaginar, que una de las particiones se cae, y existen algunos errores fuertes, no se desea escribir más en este disco. Solamente obtener todos los datos fuera de éste y retirarlos, parece ser lo adecuado. Sin embargo no es posible acceder a los archivos, ya que el sistema de archivos está dañado.
- Ahora, se desea copiar la partición entera en un archivo. Quemarlo en un CDROM, solo para nunca perderlo nuevamente. Se puede configurar un dispositivo loop, y repararlo (fsck) y tener esperanza de que pueda ser montado.
- Copiar esta partición con herramientas Un*x como *cat* o *dd* podría fallar, ya que estas herramientas abortan con algún error. dd_rescue en cambio tratará de leer y si falla, seguirá con los siguientes sectores. El archivo de salida naturalmente tendrá huecos en el, por supuesto. Se puede escribir un archivo de registro, para ver, donde se ubican todos estos errores.
- La tasa de datos perdidos es muy baja, cuando se encuentran errores. Si se interrumpe el proceso de copia, no se pierde nada. Se puede continuar en la posición siguiente. El archivo de salida será rellenado y no truncado como en otras herramientas Un*x.
- Si se tiene un espacio de sectores defectuosos dentro de la partición, podría ser una mejor idea aproximarse a este espacio desde ambos lados. La copia de dirección reversa es tu amigo.
- El tamaño de dos bloques es para optimizar en el desempeño. Tamaños de bloques grandes dan como resultado un superior desempeño, pero en caso de errores, se desea salvar cada sector único. Así que *hardbs* es mejor que sea configurado al tamaño de sector del hardware (más a menudo 512 bytes) y *softbs* a un valor grande, como el valor predeterminado 16k.

```
root@reydes: ~/tools/forensics/imaging/ddrescue-1.10
Archivo  Editar  Ver  Terminal  Ayuda
root@reydes:~/tools/forensics/imaging/ddrescue-1.10# date; ./ddrescue -b 512 /de
v/fd0 /media/hda3/tools/forensics/imaging/floppy_ddrescue.dd floppy_ddrescue.log
; date


Press Ctrl-C to interrupt
Initial status (read from logfile)
rescued:     748544 B,  errsize:    206 kB,  errors:       3
Current status
rescued:     1358 kB,  errsize:    115 kB,  current rate:        0 B/s
   ipos:     1014 kB,   errors:      28,    average rate:     1032 B/s
   opos:     1014 kB
Finished

root@reydes:~/tools/forensics/imaging/ddrescue-1.10# sha1sum ../floppy_ddrescue.
dd
ce6027744b370babf50f3e9300b3fa0e4b6178e4  ../floppy_ddrescue.dd
root@reydes:~/tools/forensics/imaging/ddrescue-1.10#
```

Figura 9. Culminación del proceso de la copia bit a bit utilizando ddrescue

```
root@reydes: ~/tools/forensics/imaging/rdd-2.0.7/src
Archivo  Editar  Ver  Terminal  Ayuda
checksumblockfilter.o  logprinter.o       rawreader.c       statsblockfilter.o
commandline.c          Makefile           rawreader.o       stdioprinter.c
commandline.h          Makefile.am        rdd.1             stdioprinter.o
commandline.o          Makefile.in        rdd-copy          strerror.c
console.c              md5blockfilter.c   rdd-copy.1        strerror.o
console.o              md5blockfilter.o   rddcopy.c         tcpwriter.c
copier.c               md5.c              rddcopy.o         tcpwriter.o
copier.h               md5.h              rdd.h             verifyblockfilter.c
copier.o               md5.o              rdd_internals.c   verifyblockfilter.o
error.c                md5streamfilter.c  rdd_internals.h   writer.c
error.h                md5streamfilter.o  rdd_internals.o   writer.h
error.o                msgdistributor.h   rddi.py           writer.o
faultyreader.c         msgprinter.c       rdd-verify        writestreamfilter.c
faultyreader.o         msgprinter.h       rdd-verify.1      writestreamfilter.o
fdreader.c             msgprinter.o       rddverify.c       zlibreader.c
fdreader.o             msgsink.h          rddverify.o       zlibreader.o
fdwriter.c             netio.c            reader.c          zlibwriter.c
fdwriter.o             netio.h            reader.h          zlibwriter.o
fileprinter.c          netio.o            reader.o
fileprinter.o          numparser.c        robustcopier.c
filereader.c           numparser.h        robustcopier.o
root@reydes:~/tools/forensics/imaging/rdd-2.0.7/src# ./rdd-copy --version
rdd version 2.0.7
root@reydes:~/tools/forensics/imaging/rdd-2.0.7/src#
```

Figura 10. Listado del directorio donde se ha compilado rdd

Instalación de dd_rescue

Para proceder a la instalación de dd_rescue desde las fuentes, se procede a descargar el archivo *dd_rescue-1.14.tar.gz*, el cual corresponde a la más reciente versión de dd_rescue. Para proceder a su instalación sólo se requiere ejecutar el comando; *make*, tal como se muestra en la Figura 6.

Utilización de dd_rescue

El modo de utilización de dd_rescue es el siguiente: *./dd_rescue [opciones] entrada salida:*

- -s ipos - Posición de inicio en el archivo de entrada (Defecto=0).
- -S opos - Posición de inicio en el archivo de salida (Defecto=ipos).
- -b softbs - Tamaño de bloque para la operación de copia (Defecto=65536).
- -B hardbs - Tamaño de bloque de retroceso en caso de errores (Defecto=512).
- -e maxerr - Sale después de *maxerr* errores (Defecto=0=infinito).
- -maxxfer - Máxima cantidad de datos a ser transmitidos (Defecto=0=infinito).
- -y syncfrq - Frecuencia de llamadas *fsync* sobre el archivo de salida (Defecto=512*softbs).
- -l archivolog - Nombre de un archivo para registrar errores y resumen (Defecto=””).
- -o archivobb - Nombre del archivo para registrar número de bloques dañados (Defecto=””).
- -r - Copia en dirección reversa (Defecto=forward).
- -t - Trunca el archivo de salida (Defecto=no).
- -w - Aborta sobre errores de escritura (Defecto=no).
- -A - Siempre escribe bloques, poniendo ceros si hay error (Defecto=no).
- -i - Interactivo: Consulta antes de sobrescribir los datos (Defecto=no).
- -f - Fuerza: Salta algunas comprobaciones de limpieza (Defecto=no).
- -p - Preserva: Preserva los propietarios / permisos (defecto=no).

Nota: los tamaños pueden ser proporcionados en unidades b(=512), k(=1024), M(=1024^2) o, G(1024^3) bytes.

Resultados de dd_rescue

El comando utilizado para realizar la copia bit a bit desde el disco flexible es:

```
./dd_rescue -l dd_rescue_floppy.err
-o dd_rescue_floppy.bb -A -b 512
/dev/fd0 tools/forensics/imaging/
floppy_dd_rescue.dd
```

dd_rescue copió 1474560 bytes del disco flexible en 783 segundos. Sus hash SHA1 son los siguientes:

```
ca52119c24df55a7b4556b21052aa2a1387e
987d  tools/forensics/imaging/floppy_
dd_rescue.dd
```

La Figura 7 muestra la culminación del proceso de copia bit a bit utilizando dd_rescue. Lo interesante de dd_rescue es la presentación de datos en tiempo real, como por ejemplo: ipos, opos, xferd, errs, que en este caso indica 256; errxfer, que en este caso indica 128.0k; y sucxfer. Finalmente la velocidad de kB/s actual y la velocidad promedio en kB/s.

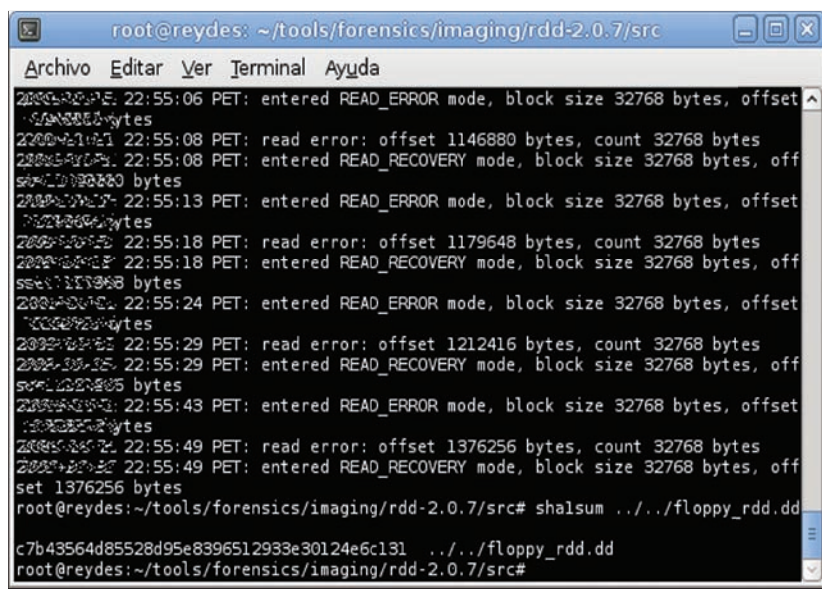


Figura 11. Culminación del proceso de la copia bit a bit utilizando rdd

GNU ddrescue

GNU ddrescue es una herramienta de recuperación de datos. Copia los datos de un archivo o un dispositivo de bloques (disco duro, CDROM, etc.) a otro, intentando rescatar los datos en caso de errores de lectura. La operación básica de ddrescue es totalmente automática. Esto es, no se tiene que esperar por la ocurrencia de un error, detener el programa, leer el archivo de registro, ejecutarlo en modo reverso, etc.

Si se utiliza la característica de archivo de registro de ddrescue, los datos son rescatados de manera muy eficiente (sólo los bloques necesarios son leídos). También se puede interrumpir el rescate en cualquier momento y resumirlo posteriormente en el mismo punto.

Combinación automática de respaldos: si se tienen dos o más copias dañadas de un archivo, CDROM, etc., y se ejecuta ddrescue en todos ellos, uno a la vez, con el mismo archivo de salida, probablemente se obtendrá un archivo completo y libre de errores. Esto es debido a que la probabilidad de tener áreas dañadas en los mismos lugares sobre diferentes archivos de entrada es muy baja. Utilizando el archivo de registro, sólo los bloques necesarios son leídos desde la segunda copia y las siguientes.

El archivo de registro es guardada periódicamente al disco. Así es que en caso de una caída se puede resumir el rescate con un pequeño recopiado. También, el mismo archivo de registro puede ser utilizado por varios comandos que copian diferentes áreas del archivo, y para varios intentos de recuperación sobre diferentes subconjuntos.

ddrescue alinea su *buffer* de Entrada/Salida al tamaño del sector de manera que pueda ser utilizado para leer dispositivos en bruto. Por razones de eficiencia, también lo alinea al tamaño de la página de memoria si el tamaño de la página es múltiplo del tamaño del sector.

Instalación de GNU ddrescue

GNU ddrescue puede ser obtenido desde los repositorios oficiales, en este caso para GNU/Linux Debian o Ubuntu, con los siguientes comandos: *apt-cache search ddrescue; apt-get install ddrescue*.

Para realizar la instalación de GNU ddrescue desde las fuentes, se procede a descargar el archivo *ddrescue-1.10.tar.gz*, el cual corresponde a la más reciente versión del programa. Sugiero la lectura del archivo *INSTALL*, que indica como seguir el procedimiento adecuado, que puede resumirse con los si-

guientes comandos; *./configure; make; make check; make install*, y de manera opcional *make install-man*. En la Figura 8 se muestra un listado de los archivos generados.

Utilización de GNU ddrescue

Su utilización es de la siguiente manera: *./ddrescue [opciones] archivoentrada archivosalida [archivolog]:*

- -b, --block-size=<bytes> Tamaño del bloque de hardware del dispositivo de entrada [512] .
- -B, --binary-prefixes Muestra multiplicadores binarios en números [Defecto SI].
- -c, --cluster-size=<bloques> Bloques de hardware para copiar a la vez [128] .
- -C, --complete-only No lee nuevos datos más allá de los límites del archivo de registro.
- -d, --direct Utiliza acceso directo al disco para el archivo de entrada.
- -D, --synchronous Utiliza escritura síncrona para el archivo de salida.
- -e, --max-errors=<n> Número máximo de áreas de errores permitidos.
- -F, --fill=<tipos> Llena tipos de áreas dados con datos de archivo de entrada. (?*/-+).
- -g, --generate-logfile Genera un archivo de registro aproximado desde una copia parcial.
- -i, --input-position=<pos> Posición de inicio en el archivo de entrada [0] .
- -m, --domain-logfile=<*archivo*> Dominio restringido a áreas marcadas como termino en *archivo*.
- -n, --no-split No intenta dividir o reintentar las áreas de error.
- -o, --output-position=<pos> Posición de inicio en el archivo de salida [ipos].
- -r, --max-retries=<n> Sale después de reintentos dados (-1=infinito) [0].
- -R, --retrim Marca todas las áreas de error como no recortadas.
- -s, --max-size=<bytes> Tamaño máximo de los datos de entrada a ser copiados.
- -S, --sparse Utiliza escrituras dispersas para el archivo de salida.
- -t --trucante Trunca el archivo de salida a un tamaño de cero.
- -T, --try-again Marca no-división, áreas no recortadas como áreas no-intentada.

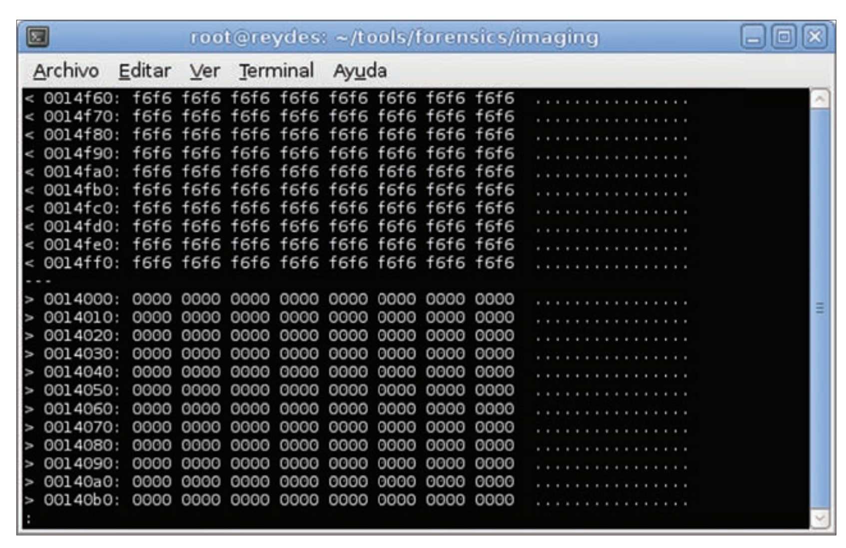


Figura 12. Parte de las diferencias entre el archivo floppy_dd.dd y floppy_dcfldd.dd

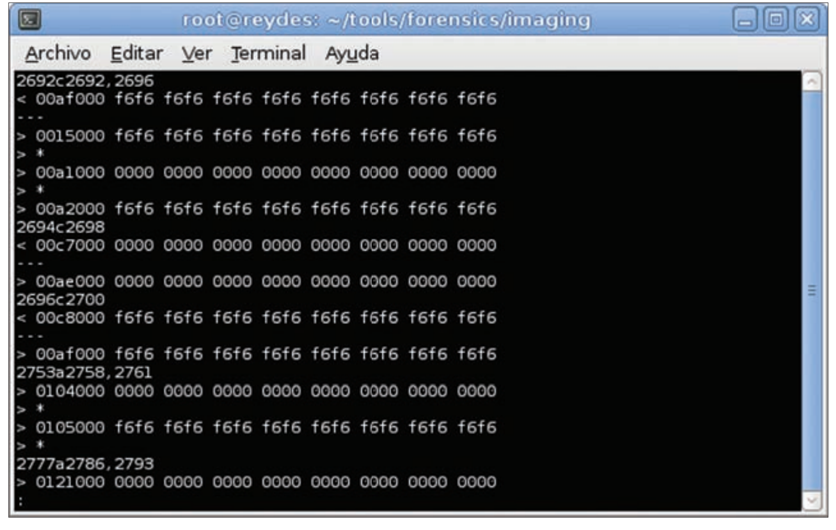


Figura 13. Parte del archivo de diferencias obtenidas con a partir de hexdump

Resultados de ddrescue

El comando utilizado para realizar la copia bit a bit desde el disco flexible es:

```
./ddrescue -b 512 /dev/fd0 tools/
forensics/imaging/floppy_ddrescue.dd
floppy_ddrescue.log
```

ddrescue copió 1474560 bytes del disco flexible en 559 segundos. Sus hash SHA1 son los siguientes:

```
ce6027744b370babf50f3e9300b3fa0e4b61
78e4  tools/forensics/imaging/floppy_
ddrescue.dd
```

La Figura 9 muestra la culminación del proceso de copia bit a bit utilizando ddrescue. Hay algunas características interesantes con la utilización de ddrescue. A diferencia de dd_rescue, ddrescue muestra en tiempo real y de manera permanente el estatus del proceso con la siguiente información: rescued, tamaño de lo rescatado en kB; errsize, tamaño de error en kB; errors, número de errores; ipos; opos; la velocidad actual y la velocidad promedio, en B/s.

A continuación se presenta la parte inicial de archivo de registro generado *floppy_ddrescue.log,* archivo que permite explotar la característica de parar el proceso de rescate y volver a ejecutarlo posteriormente donde se quedó, entre otras características ya descritas de ddrescue.

```
# Rescue Logfile. Created by GNU
ddrescue version 1.10
# current_pos  current_status
0x000F7C00     +
# pos              size
status
0x00000000  0x000AE000  +
0x000AE000  0x00001000  -
0x000AF000  0x00018000  +
```

rdd

rdd es un programa de copia forense desarrollado y utilizado por el Instituto Forense de Holanda (NFI). A diferencia de la mayoría de pro-

gramas de copia, rdd es robusto con respecto a errores de lectura, lo cual es una importante propiedad en un entorno operativo forense.

Instalación de rdd

rdd puede ser obtenido desde los repositorios oficiales, en este caso para GNU/Linux Debian o Ubuntu, con los siguientes comandos:

```
apt-cache search rdd; apt-get
install rdd
```

Para proceder a instalar rdd desde las fuentes, se procede a descargar en primera instancia el paquete de nombre *rdd-2.0.7.tar.gz*, después de desempaquetar y descomprimir el archivo, sugiero la lectura del archivo *README* e *INSTALL*. Nuevamente la forma resumida de compilar rdd es mediante los siguientes comandos: *./configure; make; make install*. La Figura 10 muestra el listado del directorio donde se ha realizado el procedimiento previamente descrito.

Utilización de rdd

La utilización de rdd es de la siguiente manera: *./rdd-copy [opciones locales] archivoentrada [archivosalida]:*

- -C, --client Ejecuta rdd como un cliente de red.
- -F, --fault-simulation <*archivo*> Simular errores de lectura especificado en <*archivo*>.
- -M, --max-read-err <*cuenta*> Abandona después de <*cuenta*> errores de lectura.
- -P, --progress <*sec*> Reporta progreso cada <*sec*> segundos.
- -S, --server Ejecuta rdd como un servidor de red.
- -b, --block-size <cuenta>[kKmMgG] Lee bloques de <cuenta> [KMG]byte a la vez.
- -c, --count <cuenta>[kKmMgG] Lee la mayoría de <cuenta> [KMG]bytes.
- -f, --force Sin piedad al sobrescribir archivos existentes.
- -i, --inetd rdd es iniciado por (x)inetd.
- -l, --log-file <*archivo*> Mensajes de registro en <*archivo*>.
- -m, --min-block-size <*cuenta*>[kKmMgK] Tamaño mínimo de bloque de lectura <*cuenta*> [KMG]byte.
- -n, --nretry <cuenta> Reintentos de lecturas fallidas <cuenta> veces.
- -o, --offset <cuenta>[kKmMgG] Salta <cuenta> [KMG] bytes de entrada.
- -p, --port <*numpuerto*> Define el puerto del servidor a <*numpuerto*>.
- -r, --raw Lee desde dispositivos en bruto (/dev/raw/raw[0-9]) .
- -s, --split <cuenta>[kKmMgG] Divide la salida, todos los archivos <cuenta> [KMG] bytes.
- -z, --compress Comprime datos enviados en la red.
- -H, --histogram <*archivo*> Almacena estadísticas derivadas de histograma en <*archivo*>.
- --md5, --md5 Calcula o imprime hash MD5.
- --sha, --sha1 Calcula o imprime hash SHA1.

Resultados de rdd

El comando utilizado para realizar la copia bit a bit desde el disco flexible es:

```
./rdd-copy -l rdd_floppy.log --sha1
/dev/fd0 tools/forensics/imaging/
floppy_rdd.dd
```

rdd copió 1474560 bytes del disco flexible en 155 segundos. Sus hash SHA1 son los siguientes:

```
c7b43564d85528d95e8396512933e3012
4e6c131  tools/forensics/imaging/
floppy_rdd.dd
```

La Figura 11 muestra la culminación del proceso de copia bit a bit utilizando rdd, en este caso muestra algunos mensajes de error y recuperación. Más adelante se muestra parte del archivo de registro con el detalle completo del proceso.

A continuación se presenta el inicio del archivo de registro *rdd_floppy.log* que ha sido especificado en la línea de comando de rdd. Entre la información importante se detalla el número de bytes escritos, el número de bytes perdidos, el número de errores de lecturas, y el hash SHA1.

```
22:55:52 -0500: === done ***
22:55:52 -0500: seconds: 155.200
22:55:52 -0500: bytes written:
1474560
22:55:52 -0500: bytes lost: 393216
22:55:52 -0500: read errors: 25
22:55:52 -0500: zero-block
substitutions: 12
```

```
Archivo  Editar  Ver  Terminal  Ayuda
diff_floppy_dd_dcfldd       floppy_dd.dd.sha1          floppy_sdd.dd.xxd
diff_floppy_dd_dcfldd_hex   floppy_dd.dd.xxd           hashdif.txt
diff_floppy_sdd_dcfldd      floppy_dd_rescue.dd        paquetes
disco_dd                    floppy_ddrescue.dd         rdd-2.0.7
disfco_dcfldd               floppy_dd_rescue.dd.sha1   sdd-1.52
floppy_dcfldd.dd            floppy_ddrescue.dd.sha1
root@reydes:~/tools/forensics/imaging# cd disco_dd/
root@reydes:~/tools/forensics/imaging/disco_dd# ls
biblio_esa                 correos               direcc.txt
buzon_raregazz_07022004    correos_asistentes    firma
clima                      correos_evento        firma_informatizate
codigo_Adsense.txt         cuenta_codigo         frase.txt
comunidades_para_textos    debian                seguridad_en_las_tecnologias2
contador_sedyer            desktop.ini           web_informatizate
root@reydes:~/tools/forensics/imaging/disco_dd# cd ..
root@reydes:~/tools/forensics/imaging# cd disco_dcfldd/
root@reydes:~/tools/forensics/imaging/disco_dcfldd# ls
biblio_esa                 correos               direcc.txt
buzon_raregazz_07022004    correos_asistentes    firma
clima                      correos_evento        firma_informatizate
codigo_Adsense.txt         cuenta_codigo         frase.txt
comunidades_para_textos    debian                seguridad_en_las_tecnologias2
contador_sedyer            desktop.ini           web_informatizate
root@reydes:~/tools/forensics/imaging/disco_dcfldd#
```

Figura 14. Listados de los directorios donde se han montado las imágenes obtenidas con dd y dcfldd

```
Archivo  Editar  Ver  Terminal  Ayuda
/media/hda3/tools/forensics/imaging/disco_dcfldd/frase.txt
/media/hda3/tools/forensics/imaging/disco_dcfldd/seguridad_en_las_tecnologias2
root@reydes:~/tools/forensics/imaging/disco_dcfldd# md5deep -l -r -m ../discoddl
ista *
biblio_esa
buzon_raregazz_07022004
clima
codigo_Adsense.txt
comunidades_para_textos
contador_sedyer
correos
correos_asistentes
correos_evento
cuenta_codigo
debian
desktop.ini
direcc.txt
firma
firma_informatizate
frase.txt
seguridad_en_las_tecnologias2
root@reydes:~/tools/forensics/imaging/disco_dcfldd# md5deep -l -r -x ../discoddl
ista *
root@reydes:~/tools/forensics/imaging/disco_dcfldd#
```

Figura 15. Comparación de los hash MD5 de los archivos existentes en ambas imágenes bit a bit

```
22:55:52 -0500: MD5: <none>
22:55:52 -0500: SHA-1: c7b43564d8552
8d95e8396512933e30124e6c131
```

Resultados parciales

Los resultados obtenidos al aplicar todas las herramientas incluidas para obtener una imagen bit a bit del floppy disk se pueden resumir en la Tabla 1.

Ahora la pregunta a responder es la razón por la cual todos los hash SHA1 obtenidos son diferentes. En primera instancia se pueden exponer las siguientes razones:

- Es un disco flexible, como se ha mencionado al inicio del presente artículo data aproximadamente del mes de enero del año 2005, y no ha sido guardado en unas condiciones aceptables, llámese polvo, suciedad y humedad lo cual afecta la propia integridad del medio de almacenamiento y su proceso de lectura.
- En el caso de la lectora de discos, aunque no ha sido muy utilizada, ya tiene aproximadamente 4 años. Aquí estamos hablando de un dispositivo mecánico que puede tener algunas variables al realizar al proceso de lectura del dispositivo.
- Los programas seleccionados pueden permitir la utilización de diferentes parámetros u opciones que trabajan de manera particular al encontrar algún problema en el proceso de lectura del medio. Esto afecta obviamente los datos que pueden ser obtenidos desde el medio.

Evaluaciones Finales

Bien, la curiosidad es un atributo importante para temas relacionados al cómputo forense. Por lo tanto, a modo de ejercicio, se procede a ubicar algunas de las zonas diferentes entre el archivo obtenido con dd *floppy_dd.dd.xxd*, y el archivo obtenido con dcfldd, *floppy_dcfldd.dd*.

Para este proceso se utiliza el programa xxd, el cual crea un volcado hexadecimal de un archivo o entrada estándar. Y también el comando diff que compara archivos línea por línea. La Figura 12 muestra parte de estas diferencias.

```
# xxd floppy_dd.dd > floppy_dd.dd.xxd
# xxd floppy_dcfldd.dd > floppy_
dcfldd.dd.xxd
# diff floppy_dd.dd.xxd floppy_
dcfldd.dd.xxd > diff_floppy_dd_dcfldd
```

Este proceso puede aplicarse a todos los archivos al mismo tiempo.

Otra herramienta que podemos utilizar en este proceso es hexdump que realiza volcados ASCII, decimales, hexadecimales y octales. Nuevamente se ha procedido a generar archivos de volcados con hexdump para los archivos *floppy_dd.dd* y *floppy_dcfldd.dd* y luego obtener sus diferencias con el resultado del comando diff redireccionado a un archivo. La Figura 13 muestra parte del archivo que contiene los resultados del proceso descrito.

Las diferencias obtenidas con los códigos hexadecimales 00 y F6 en ambos archivos, requieren de un análisis más cuidadoso, pero se puede coligar que corresponde a la manera en que dd y dcfldd han manejado las zonas del disco flexible donde se ha presentado algún problema con la lectura de datos. Recordar que solamente estamos tomando como ejemplo a los resultados de dd y dcfldd.

Seguimos con la prueba, se procede a montar las dos imágenes del ejemplo con los siguientes comandos:

```
# mount floppy_dd.dd disco_dd/ -o
ro,loop,nodev,noexec
# mount floppy_dcfldd.dd disco_dcfldd/
-o ro,loop,nodev,noexec
```

El resultado del montaje, y el listado del contenido de ambas imágenes, se puede visualizar en la Figura 14.

El propósito de esta prueba es tratar de obtener alguna diferencia entre los archivos existentes, es decir que no han sido eliminados, de las réplicas bit a bit obtenidas con dd y dcfldd. Para esto se utilizará el comando md5deep, el cual es un conjunto de programas para múltiples plataformas que calculan mensajes resumen MD5, SHA-1, SHA256, Tiger o Whirpool de un número arbitrario de archivos. md5deep es similar al programa md5sum encontrado en el paquete *GNU Core Utilities*, pero tiene algunas características adicionales.

En primera instancia se ha procedido a obtener un hash de todos los archivos residentes de la imagen obtenida con dd, con el siguiente comando:

```
# cd disco_dd/
# md5deep -b -r * > ../discoddlista
```

La opción -r habilita el modo recursivo. Es decir se recorren todos los directorios, la opción -b no incluye información de los directorios en los nombres de archivo mostrados. Ahora se procede a ingresar al directorio donde residen los archivos existentes de la imagen obtenida con dcfldd.

```
# cd disco_dcfldd/
# md5deep -l -r -m ../discoddlista *
```

En este caso las opciones de md5deep son: -l habilita las rutas relativas de los archivos, en lugar de imprimir las rutas absolutas de cada archivo; la opción -m, habilita el modo de correspondencia, el archivo indicado debe ser una lista de hash conocidos. También se puede utilizar la opción -x que tiene una función inversa a la opción -m que muestra las coincidencias negativas.

```
# md5deep -l -r -x ../discoddlista *
```

La Figura 15 muestra los resultados de estos dos procesos descritos. Se debe recordar que en todos estos procesos detallados se ha trabajado con hash MD5, para utilizar hash SHA1 se utiliza simplemente el comando sha1deep. Los resultados obtenidos no muestran ninguna diferencia entre los archivos. El mismo proceso aplicado a las demás imágenes obtenidas tampoco muestran diferencia alguna a este nivel.

Hasta este punto también se recordará una máxima del cómputo forense, la cual es la utilización de dos o más herramientas de análisis, con el propósito de comparar resultados y no confiarse de los resultados obtenidos con una única herramienta. Y de manera consecuente llega la pregunta, ¿cómo se realiza la sustentación de los seis diferentes hash obtenidos por las seis herramientas?

En primera instancia recordar otra máxima del cómputo forense, la cual es la documentación, se debe documentar todo el proceso realizado con las herramientas, desde su compilación, configuración y aplicación al obtener la evidencia de las unidades de almacenamiento, dado que este proceso de ser requerido, necesitará ser sustentado ante una solicitud.

En este caso se ha obviado la utilización de un bloqueador de escritura, pero se ha utilizado la "protección" física que tiene de manera inherente un disco flexible, como es el caso. Siempre, repito, siempre se debe de utilizar un bloqueador de escritura durante la copia bit a bit de la evidencia.

Ahora bien, una manera de apoyar nuestra correcta metodología al realizar la captura de la evidencia, compilar las herramientas y realizar la réplica bit a bit de los medios, es la utilización de un hash Difuso. Para este propósito se utiliza ssdeep, el cual es un programa para calcular y corresponder una porción de hash disparado por contexto (CTPH), también denominado hash difuso. Tales entradas tienen secuencias de bytes idénticos en el mismo orden, sin embargo los bytes entre estas secuencias pueden ser diferentes en contenidos y longitud.

Como se puede apreciar en la Figura 16, al aplicar el comando ssdeep a las seis imágenes obtenidas con las seis herramientas expuestas en el presente artículo todos los hash difusos obtenidos son idénticos, lo cual puede ser de gran apoyo a la metodología utilizada. Se pueden utilizar los siguientes comandos para automatizar el proceso de comparación:

```
# ssdeep -b *.dd > hashdif.txt
```

La opción -b no incluye información del directorio de los nombres de archivo.

```
# ssdeep -bm hashdif.txt *.dd
```

La opción -m <archivo> carga el archivo de hashes conocidos para ser utilizados en la correspondencia. Este archivo debe ser una salida previa del programa y tener la cabecera correcta. Y muestra solamente aquellos archivos que corresponden con archivos conocidos y qué archivos corresponden con ellos. Aunque los nombres de archivo no deben contener caracteres UNICODE, se pueden manejar hashes con nombre de archivos UNICODE.

El tema del hash difuso puede ser un buen motivo para redactar un futuro artículo de cómputo forense.

Conclusiones

Existen en la actualidad excelentes herramientas forenses para la adquisición, captura u obtención de la evidencia, las cuales se incluyen en la mayoría de Live CDs basados en GNU/Linux. Estos Live CDs permiten una adecuada utilización de estas herramientas para realizar una copia bit a bit de los medios objeto del análisis.

Solamente una adecuada metodología, un buen nivel de conocimiento técnico y buenas prácticas permiten sustentar los resultados que se obtienen durante la fase del análisis de la evidencia. En este caso puntual se utilizaron otras herramientas y técnicas para sustentar los diferentes resultados obtenidos con la utilización de estas seis herramientas de adquisición de evidencia.

Recordar nuevamente algunas máximas del cómputo forense: documentar absolutamente todos los procedimientos, utilizar dos o más herramientas para realizar el análisis y validar los resultados obtenidos y presentar estos resultados de manera que sea fácil de comprender. Hasta una próxima oportunidad.

```
root@reydes: ~/tools/forensics/imaging
Archivo Editar Ver Terminal Ayuda
root@reydes:~/tools/forensics/imaging# ls -l *.dd
-rw-r--r-- 1 root root 1474560 2009-10-15 21:08 floppy_dcfldd.dd
-rw-r--r-- 1 root root 1474560 2009-10-16 08:49 floppy_dd.dd
-rw-r----- 1 root root 1474560 2009-10-15 22:14 floppy_dd_rescue.dd
-rw-r--r-- 1 root root 1474560 2009-10-15 22:41 floppy_ddrescue.dd
-r--r--r-- 1 root root 1474560 2009-10-15 22:55 floppy_rdd.dd
-rw-r--r-- 1 root root 1474560 2009-10-15 21:52 floppy_sdd.dd
root@reydes:~/tools/forensics/imaging# ssdeep
ssdeep: No input files
root@reydes:~/tools/forensics/imaging# ssdeep *.dd
ssdeep,1.0--blocksize:hash:hash,filename
768:pbH+WbO/20pu1XGhohetoC1INfXRssQGoKY+:5B90puODV1INfR2GoT+,"/media/hda3/tools/
forensics/imaging/floppy_dcfldd.dd"
768:pbH+WbO/20pu1XGhohetoC1INfXRssQGoKY+:5B90puODV1INfR2GoT+,"/media/hda3/tools/
forensics/imaging/floppy_dd.dd"
768:pbH+WbO/20pu1XGhohetoC1INfXRssQGoKY+:5B90puODV1INfR2GoT+,"/media/hda3/tools/
forensics/imaging/floppy_dd_rescue.dd"
768:pbH+WbO/20pu1XGhohetoC1INfXRssQGoKY+:5B90puODV1INfR2GoT+,"/media/hda3/tools/
forensics/imaging/floppy_ddrescue.dd"
768:pbH+WbO/20pu1XGhohetoC1INfXRssQGoKY+:5B90puODV1INfR2GoT+,"/media/hda3/tools/
forensics/imaging/floppy_rdd.dd"
768:pbH+WbO/20pu1XGhohetoC1INfXRssQGoKY+:5B90puODV1INfR2GoT+,"/media/hda3/tools/
forensics/imaging/floppy_sdd.dd"
root@reydes:~/tools/forensics/imaging#
```

Figura 16. Utilización de ssdeep a las seis imágenes obtenidas del mismo disco flexible



En la red

- Forensics Wiki - *http://www.forensicswiki.org/*
- Floppy Disk - *http://en.wikipedia.org/wiki/Floppy_disk*
- dd - *ftp://ftp.gnu.org/gnu/coreutils/*
- dd (Unix) - *http://en.wikipedia.org/wiki/Dd_(Unix)*
- dcfldd - *http://dcfldd.sourceforge.net/*
- sdd - *http://freshmeat.net/projects/sdd/*
- dd_rescue - *http://www.garloff.de/kurt/linux/ddrescue/*
- ddrescue - *http://www.gnu.org/software/ddrescue/ddrescue.html*
- rdd - *http://rdd.sourceforge.net/*
- aimage - *http://www.afflib.org/*
- libewf - *http://libewf.sourceforge.net/*
- ssdeep - *http://ssdeep.sourceforge.net/*
- md5deep - *http://md5deep.sourceforge.net/*



Sobre el autor

Alonso Eduardo Caballero Quezada es Brainbench Certified Computer Forensics (U.S.) y GIAC SSP-CNSA. Actualmente trabaja como consultor en Hacking Ético y Cómputo Forense. Perteneció por muchos años al grupo RareGaZz. Actualmente es integrante del Grupo Peruano de Seguridad PeruSEC. Se presenta de manera frecuente en cursos y ponencias, las cuales se enfocan en Cómputo Forense, Hacking Ético, Análisis de Vulnerabilidades, Pruebas de Penetración, GNU/Linux y Software Libre. Su correo electrónico es ReYDeS@gmail.com y su página personal está en: *http://www.ReYDeS.com.*

Hosting: Cloud VS Clúster

Tobia Caneschi, Jefe de Investigación y Desarrollo en el Departamento Hosting & Server de Nominalia.com

Desde hace algunos meses, en todos los sitios del sector, foros y comunidades de profesionales no se habla más que de Cloud Hosting o computación en nube. Pero ¿cuáles son las ventajas reales de utilizar esta nueva tecnología? Y, sobre todo, ¿para qué tipo de necesidades puede ser la solución perfecta?

Empezaremos por analizar las características positivas y negativas del Cloud Hosting comparándolo con el servicio más difundido y contrastado del mercado, el Managed Hosting o Hosting en Clúster.

Múltiples empresas nacionales e internacionales tienen ofertas de hosting de este tipo. La solución se basa en el concepto de ofrecer a los usuarios, de manera totalmente gestionada, una parte de un espacio compartido con la posibilidad de utilizar scripts, bases de datos y servicio web, además de FTP, SSH, SSL, etc.

Estas soluciones clásicas de hosting compartido se diferencian entre sí sobre todo por la arquitectura que las aloja y por las soluciones utilizadas para gestionar la plataforma, que pueden ser propias y desarrolladas por sus creadores o bien "llave en mano", como CPanel o Hsphere.

Las ofertas más avanzadas están basadas en una arquitectura en clúster con balanceo de carga (load balancing) y almacenamiento compartido a nivel enterprise, es decir, sin point of failure, y un tiempo de uptime muy alto. Eligiendo una de estas soluciones en clúster, los usuarios podrán disponer de una estructura a la que jamás tendrían acceso a nivel individual debido a los altos costes de tal estructura.

El Cloud Computing es el último paso de la virtualización, que permite aprovechar como "motor" ya no un servidor único y limitado, sino un conjunto de servidores. El estadio de evolución de esta tecnología está suficientemente avanzado como para permitir subir cualquier imagen servidor y hacer que esté siempre activa, sin downtimes, gracias justamente a la distribución de los recursos que realiza el cloud computing.

El sistema se ocupa de gestionar los problemas de hardware de los distintos servidores de manera transparente, permitiendo trasladar el servidor hacia otros recursos hardware "en caliente" y en pocos segundos.

¿Cuál es entonces la mejor solución?

Para necesidades de un cliente medio, si comparamos una solución de Managed Hosting pre-establecida y no flexible con una solución parecida de Cloud Computing, podremos afirmar que ésta última ofrece a los usuarios ventajas y prestaciones muy parecidas.

¿Y en cuanto a la flexibilidad?

Una solución Cloud permite, incluso a nivel de un solo cliente, pasar y evolucionar en todo momento y sin límites de una solución "manager" a otro tipo de servicio. En la solución de hosting en clúster, ésta mantendrá su configuración inicial y, probablemente, no podrá satisfacer las necesidades del cliente cuando éstas cambien.

La diferencia primordial no está en el hecho de compartir los recursos, ya que también un Hosting Cloud funciona con recursos no dedicados. Con respecto a un servicio que funciona sobre un servidor físico, la virtualización sobre la que se basa el Cloud Hosting incluye más pasos que el hosting en Clúster , lo cual, evidentemente, no puede garantizar mayor velocidad.

Conclusiones

Para concluir este análisis, podríamos decir que la tecnología del Cloud Computing es, con toda certeza, un paso decisivo para el desarrollo de servicios "siempre activos", garantizando up-times muy altos, y dejando posible la expansión. Pero ¿para qué usuarios es la mejor solución?

Las ofertas de computación en nube aún no han demostrado estar por delante de tecnologías más sólidas y consolidadas como un clúster de servidores. En general, tienen un coste mayor con respecto a las soluciones de hosting compartido.

Con presupuestos medianos-altos, la solución del cloud hosting resulta muy adecuada. Una solución cloud con stack completo presenta la posibilidad de cambiar el servicio de acuerdo con las necesidades de la empresa, por ejemplo, añadir servicios a su propia parte de "nube", como un servlet java, o bien desarrollar una aplicación propia del server o instalar una de terceras partes, etc. Sin embargo, para usuarios con presupuestos bajos, necesidad de rendir y, sobre todo, con necesidades de gestión muy variadas, aconsejamos sin duda una solución de tecnología clúster, y no cloud.

La principal característica que hay que tener siempre en cuenta cuando se elige un servicio de Managed Hosting es la relación entre usuarios y recursos utilizados. Si encuentra una empresa que le ofrece este tipo de información, ya sea cloud o clúster, podrá tener la certeza de que su hosting o servicio rendirá siempre al máximo sin sufrir deterioros. En definitiva, lo importante no es elegir una solución u otra, sino confiar en un proveedor con experiencia que sea capaz de ofrecerle la mejor arquitectura posible con la mejor configuración y que más rinda.

Para todo el resto, suerte en vuestra búsqueda y, si desea profundizar en el tema, no dude en consultar los enlaces para obtener más información.

Sobre Nominalia

Nominalia forma parte de la sección Dada.pro, el departamento que se ocupa de los servicios orientados a los negocios de las empresas y de clientes profesionales del Grupo Dada. El Grupo Dada es una compañía líder en la gestión de la presencia en Internet de personas y empresas, y tiene sedes en Italia, Reino Unido, Francia, Portugal y Holanda a través de las marcas Register.it, Namesco Limited y el Grupo Amen.

Sobre el autor

Tobia Caneschi es co-creador de la plataforma de Hosting Compartido en Clúster / Jefe de Investigación y Desarrollo en el Departamento Hosting & Server de Nominalia.com / Register.it

Máster de la Comunidad Europea de Programación en ambiente Solaris, en sistemas Sun, redes y sistemas operativos.

Analisis de soportes
de datos con herramientas de código libre

Francisco Lázaro

Hace algunos años dos estudiantes del M.I.T., Simson L. Garfinkel y Abi Shelat, compraron alrededor de 150 discos duros procedentes de subastas de Internet, empresas que querían renovar sus equipos y otras fuentes. Su objetivo consistía en realizar un estudio sobre la información que dejan los usuarios en los discos duros después de desprenderse de ellos. La investigación puso de manifiesto que gran parte de los mismos contienen datos sensibles pertenecientes a sus antiguos dueños. En uno de los soportes, con la ayuda de un script de Perl para rastrear expresiones regulares, los dos investigadores encontraron miles de números de tarjetas de crédito.



linux@software.com.pl

La prueba se ha repetido en diferentes países del mundo con similares resultados y unas conclusiones decepcionantes: el mercado de ocasión para ordenadores, discos duros, llaves USB, iPods y tarjetas de memoria está lleno de datos personales e información confidencial que, vendida como quien dice a peso, va cambiando de manos sin ningún control. La terca persistencia del dato digital y nuestra incapacidad para eliminarlo cuando ha dejado de hacer falta constituyen el signo trágico de los tiempos. No solo los arqueólogos del futuro lo tienen fácil: también quienes se dedican en el presente al voyeurismo informático y a la utilización indebida de datos. El tamaño de este bazar es enorme: todos los años salen al mercado varios cientos de millones de discos duros usados.

Tecnología forense: del gurú al geek

Durante las décadas de los 70 y los 80 la Informática Forense constituía una especialidad misteriosa y solo al alcance de unos pocos iniciados que conocían los sistemas a fondo. En ocasiones se trataba de los mismos que los habían desarrollado. Con los primeros PC tuvo lugar un avance decisivo en la historia de la democracia digital: el programa undelete, los editores hexadecimales y las primeras versiones de las Utilidades Norton hicieron posible que el usuario de a pie pudiese rastrear virus, recuperar archivos borrados y familiarizarse con las particularidades internas del sistema. En una época en la que no había mucho para elegir aparte de MS DOS, CPM, Minix y OS2, sin darse cuenta estaba ya aprendiendo sus primeras lecciones de tecnología forense.

Tras el triunfo arrollador de Windows y los entornos gráficos a mediados de los 90 aparecerían sofisticados y caros programas como EnCase y FTK, capaces de indexar toda la evidencia en un soporte de datos y dirigidos a un mercado elitista compuesto por la administración de Justicia, las Fuerzas de Seguridad y los departamentos de IT de las grandes empresas. Casi de manera simultánea, la explosión cámbrica de Linux y la cultura del software libre pusieron al alcance del usuario unas potentes herramientas que hasta entonces habían sido exclusivas de los entornos Unix. En las páginas que siguen vamos a hablar

```
C:\>dir/w

 Volume in drive C is DISNORTE
 Volume Serial Number is 1333-10D5
 Directory of C:\

[COFRADIA]      [TEMP]          MOVICOMM.BAT    [redacted]      [redacted]
[PERSONAL]      TREEINFO.NCD    [DBASE]         [DOS]           [!2DISIS]
[NORTON]        [ORDAGO]        [!2DICON]       [SYMPHONY]      [WINDOWS]
[!!DC_001]      DI.BAT          COMMAND.COM     AUTOEXEC.COI    CONFIG.COI
D.BAT           AD.BAT          GMOUSE.COM      MODEM.BAT       VT100.DOC
VT100.MEF       WINA20.386      CONFIG.OLD      AUTOEXEC.WON    INSTADU.BAT
INSTALAR.BAT    CONTA.BAT       CONFIG.SYS      AUTOEXEC.BAK    [LUW]
BOT.SPL         [CONTAWIN]      [MOUSE]         AUTOEXEC.BIT    [ST6220]
AUTOEXEC.BFX    [WINWORD]       AUTOEXEC.BAT    [BFWIN]         [BC1WIN]
       45 file(s)        135,787 bytes
                     102,195,200 bytes free

C:\>_
```

Figura 1. Arranque de un disco antiguo con ms-dos bajo vmware

de algunas de ellas, así como de su utilidad para la investigación de soportes de datos informáticos.

Material de partida

Los primeros discos duros utilizados en el ensayo de software forense proceden principalmente del entorno: un ordenador desguazado en casa o en la empresa, discos viejos comprados en el rastro o en tiendas de informática, etc. También conseguimos algunos en los pasillos de un prestigioso centro universitario donde se había declarado una retirada masiva de equipos por el método expeditivo y lamentable -desde el punto de vista de la seguridad- de sacarlos a los pasillos para que fueran recogidos por una empresa de reciclajes. Ni que decir tiene que numerosos estudiantes, curiosos e incluso gente ajena a la institución se adelantaron al servicio de recogida, consiguiendo así acceso inmediato al hardware. Sin comentarios.

Otra opción es conseguirlos en eBay, donde en todo momento hay un desfile constante de ejemplares aguardando a que alguien puje por ellos. De paso compramos alguna que otra llave USB de segunda mano y media docena de tarjetas de memoria SD para cámara digital.

Herramientas

Las posibilidades que ofrece Linux para la investigación forense son prácticamente ilimitadas, con herramientas de gran calidad que no tienen nada que envidiar al software utilizado por la Administración y las Fuerzas del Orden Público. Nos limitaremos a emplear solamente algunas de ellas, potentes y de manejo sencillo. Para ser investigador forense no es imprescindible poseer un laboratorio bien equipado y ser titular de caras licencias de software, sino tener claridad mental, rigor en el método y objetivos concretos. El resto nos lo proporciona ese gran árbol de la vida tecnológica que es el mundo del software libre, siempre generoso con el usuario dispuesto a trabajar.

Esta es la caja de herramientas utilizadas con Kubuntu 8.04:

- Gparted,
- dd y dd_rescue,
- TSK - The SleuthKit,
- Photorec.

Acceso al soporte de datos

Lo primero que hacemos es conectar el soporte a un interfaz adecuado de nuestra estación de trabajo. En el caso que nos ocupa se trata exclusivamente de discos duros ATA, ya anticuados para lo que se suele considerar normal en nuestros días, con capacidades que van desde los 200 MB hasta los 40 GB. Casi todos los ordenadores actuales llevan discos SATA. Sin embargo es habitual que hasta las placas más modernas incluyan un conector IDE, el cual nos servirá para enganchar nuestros discos al sistema. Si entre los discos duros de segunda mano hubiera algún SCSI habrá que conseguir también una tarjeta para conectarlo al ordenador.

Nuestra máquina se ha quedado anticuada, aunque sirve de sobra para lo que queremos: un Athlon con BIOS de 2002 y 512 MB de memoria RAM, arranque dual (Windows XP/Kubuntu 8.04), dos discos duros conectados en tándem master-slave en el primer IDE (/dev/sda de 160 GB para los sistemas operativos y /dev/sdb de 320 GB para guardar las imágenes forenses), grabadora de DVD/CD pinchada en el master del segundo IDE (/dev/sdc) y el conector restante disponible para añadir cada uno de los soportes investigados (/dev/sdd), previa configuración adecuada de los jumpers en posición slave (esclavo). Obsérvese que estamos trabajando con Ubuntu, en caso de utilizar otra distribución de Linux los dispositivos pueden llevar nombres diferentes: /dev/hda, /dev/hdb, /dev/hdc y /dev/hdd. Las tarjetas de memoria SD serán leídas a través de un adaptador USB. También se puede acceder a los discos IDE a través de un adaptador USB. En tal caso los jumpers de las unidades irán por lo general colocados en posición master (maestro).

Conviene decir que ésta no es una forma limpia de hacer las cosas. Tampoco una buena idea si de lo que se trata es de conseguir pruebas y elaborar informes periciales para el juzgado. El tratamiento forense de la evidencia con fines judiciales requiere mayores cuidados, sobre todo para evitar que alguien la impugne con el pretexto de que ha podido ser manipulada. La conexión debe establecerse mediante dispositivos hardware que permitan el acceso al soporte en modo estricto de solo lectura, o como mínimo arrancando el ordenador con una distribución LINUX de la cual

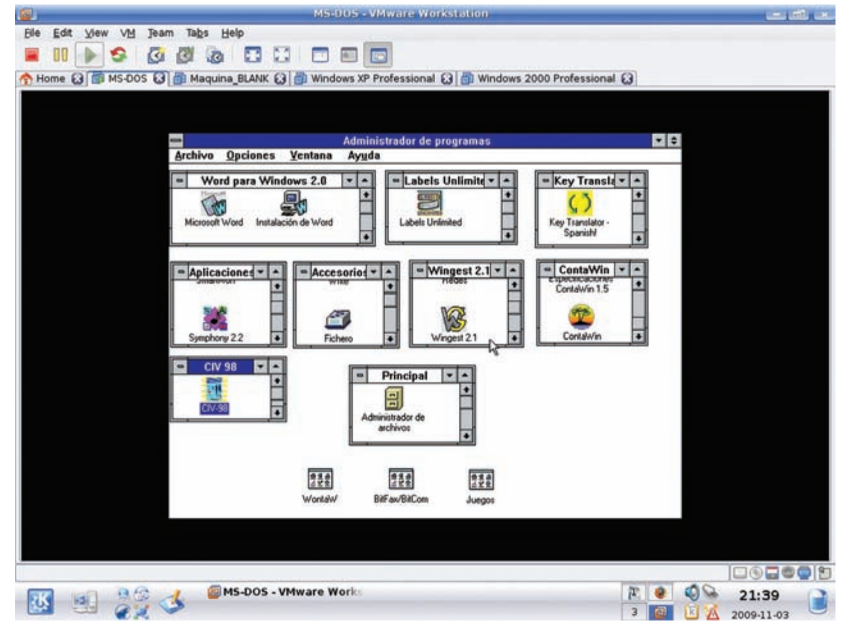


Figura 2. Disco duro antiguo con Windows 3.1 funcionando bajo vmware

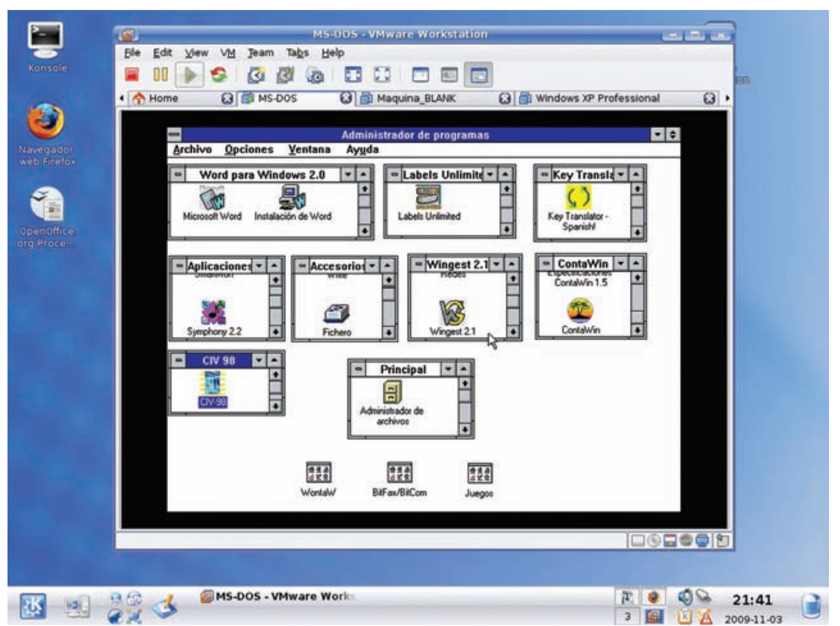


Figura 3. Imagen virtual de un disco duro con ms-dos y windows 3.1 funcionando bajo vmware

estemos seguros que no va a alterar ni un solo bit de la evidencia. Aunque no es frecuente hallar contadores de journaling en los discos duros de hace más de cinco años, nunca se sabe con qué nos podemos encontrar cada vez que conectamos un dispositivo desconocido a nuestra estación de trabajo.

Ahora no estamos trabajando para el fiscal. Hemos venido a hacer minería de datos, un poco a lo bruto, y podemos tolerar algo de contaminación en el entorno. El primer examen lo llevamos a cabo mediante el gestor de particiones Gparted. Si no está disponible, lo instalamos tecleando lo siguiente en una consola bash:

```
local@local-desktop:~$ sudo apt-get
install gparted
```

Para iniciar *gparted* podemos ir al menú de aplicaciones (en Sistema) o llamarlo desde la línea de comando. En el interfaz del programa seleccionamos el dispositivo conectado -que en la configuración a la cual se acaba de hacer referencia, como slave del segundo interfaz IDE, se encuentra en */dev/sdd-* , para saber si está particionado, de qué tipo de particiones se trata (FAT16, FAT32, NTFS, ext2, etc.) y si existen datos dentro del dispositivo. Si es así podremos acceder a ellos sin más que montar la partición y examinar su contenido desplazándonos entre los directorios mediante comandos de consola, un navegador de archivos o bien con 'mc', la versión en Linux del primitivo Comandante Norton. Cuando se trata de una partición FAT los archivos borrados se pueden listar con Sleuthkit:

```
local@local-desktop:~$ fls -rd /dev/
sdd1
```

Y para rescatar un archivo borrado, por ejemplo un documento de MS-Word:

```
local@local-desktop:~$ icat /
dev/sdd1 [número de inode] >
evidencia.doc
```

Imágenes en bitstream

Teniendo espacio suficiente en nuestra estación es mejor realizar una imagen del soporte para trabajar sobre ella y no sobre el dispositivo físico. Las razones son obvias: no queremos modificar el contenido del disco duro ni someter a sus cabezas de lectura al esfuerzo mecánico que precisa el rastreo exhaustivo de datos. Estamos manipulando discos antiguos cuyo historial de uso se desconoce. Ignoramos cómo les fue en su vida anterior, si han recibido golpes o sacudidas a consecuencia de las cuales pudieran tener sectores defectuosos y otros daños. Más de uno se hallará al final de su vida útil y no nos interesa que muera en plena operación de búsqueda. De modo que en el dispositivo de almacenamiento -es decir, el segundo disco duro de nuestra estación de trabajo forense- creamos un directorio para las imágenes, subimos hasta él y adquirimos el soporte íntegro, con todas sus particiones, el espacio no asignado y el sector de arranque:

```
local@local-desktop:~/mnt/sdb1$
mkdir imagenes_forenses

local@local-desktop:~/mnt/sdb1$ cd
imagenes_forenses

local@local-desktop:~/mnt/sdb1/
imagenes_forenses$ dd if=/dev/sdd
of=imagen_disco1.dd
```

El comando dd dispone de opciones que nos permiten optimizar el proceso de copia en bitstream. Si esto fuera prioritario en la investigación podríamos servirnos de ellas, por ejemplo definiendo el tamaño de los bloques de datos mediante el parámetro 'bs'. Haremos lo mismo con cada uno de los discos duros -dependiendo siempre del espacio disponible-

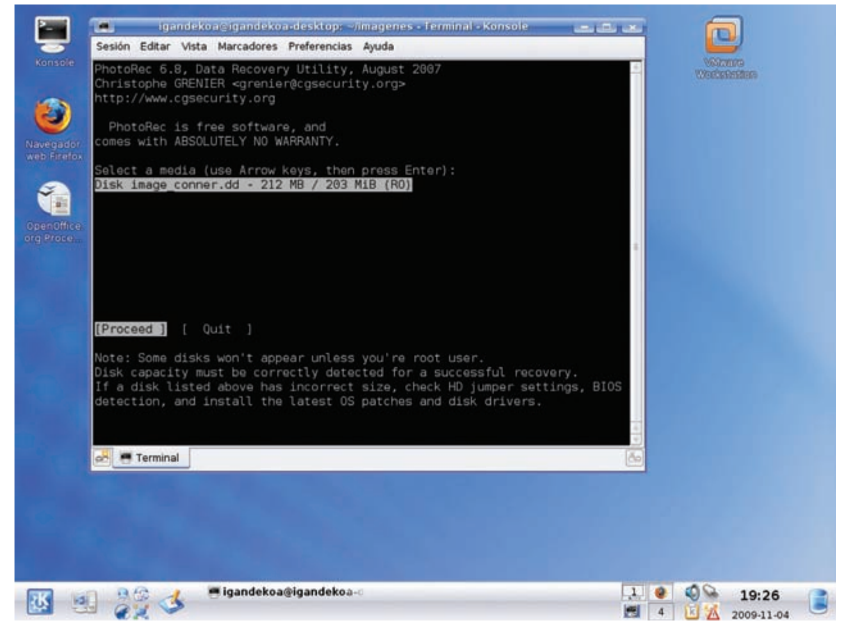


Figura 4. Interfaz de photorec

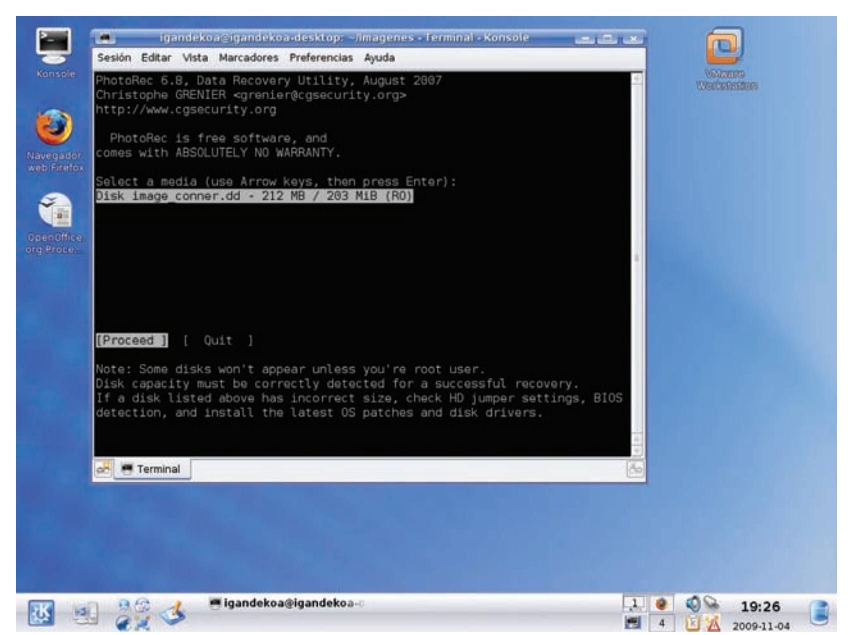


Figura 5. Interfaz de photorec 2

y con las tarjetas SD. Si los soportes de datos tienen sectores defectuosos, 'dd' abortará sin haber completado su misión. Por fortuna disponemos de 'dd_rescue' y otras herramientas similares diseñadas para recuperar datos de discos defectuosos, capaces de realizar una copia en bitstream saltándose los sectores ilegibles. Perderemos los datos grabados en las partes dañadas del disco, pero a cambio tendremos acceso a los que pueden ser recuperados sin problema.

Alternativamente a 'dd_rescue' se puede utilizar el propio 'dd' con la opción "bs=512 conv=noerror,sync", aunque no resulta recomendable debido al tiempo que consume y a los movimientos repetidos que obliga a hacer a las cabezas de lectura sobre sectores dañados. El disco podría estar en las últimas y no conviene forzarlo mecánicamente. Volviendo a dd_rescue, existe un script bash llamado dd_rhelp que maneja este útil programa -escrito en C por Kurt Garloff- de modo que las partes sanas del dwisco duro se recuperan antes de comenzar las operaciones de lectura sobre los sectores defectuosos, ahorrando tiempo al investigador y minimizando la pérdida de datos en caso de un fallo catastrófico.

Recuperando archivos con Photorec

Borrar archivos no significa eliminarlos por completo, pudiéndose recuperar mediante técnicas adecuadas. Aunque este hecho es conocido, mucha gente se sorprende al saber que la pérdida de datos no es definitiva ni siquiera después de haber formateado el disco duro. Una vez más nos encontramos ante una reminiscencia mítica de los primeros tiempos del PC, cuando el formateo accidental era lo peor que podía suceder. El comando "format c:" imponía respeto. Solamente los muy entendidos se atrevían a teclearlo, plantando cara a ese mensaje con signos de admiración que en todos los sistemas advierte de una pérdida total de los datos.

En realidad formatear un disco significa borrar todas las estructuras de datos del sistema de archivos creando otras nuevas encima de ellas. Los datos efectivamente se pierden, más en sentido literal que en el de eliminación física, puesto que a pesar de que la mayor parte continúan estando en el disco duro, resulta imposible llegar hasta ellos al no existir ya el sistema de archivos. No olvidemos que éste, en última instancia, no es más que una base de datos que permite gestionar y mantener los archivos de un volumen. Si la eliminamos nos resultará imposible hacerlo. Ni siquiera sabremos dónde están.

En un disco reformateado las herramientas de *The Sleuth Kit* no detectan ningún archivo, puesto que la FAT (File Allocation Table) o la MFT (Master File Table), si se trata de particiones Windows), o la tabla de inodes (en Linux), han sido reinicializadas. Incluso el más potente software de investigación utilizado por la policía no podrá hacer otra cosa que inventariar el disco a un nivel elemental: sectores, clusters o porciones del volumen convertidas en archivos para hacer búsquedas de caracteres. Pero ni rastro de los archivos y carpetas originales. Para llegar hasta ellos es preciso utilizar herramientas de *file carving*.

En un artículo publicado en esta revista (Nº. 59 - Alonso Eduardo Caballero Quezada: *"Foremost y Scalpel: Herramientas de recuperación de archivos"*) se exponen los principios del "file carving" (tallado de archivos) y el funcionamiento de dos herramientas imprescindibles para esta labor, con algoritmos de búsqueda que van más allá de la simple comparación de cabeceras y pies con una base de datos, extrayendo archivos en función de sus estructuras internas de datos. Aquí nos vamos a servir de una herramienta más sencilla: *Photorec*. Pese a que su nombre sugiere una especialización gráfica, también recupera archivos de otros tipos.

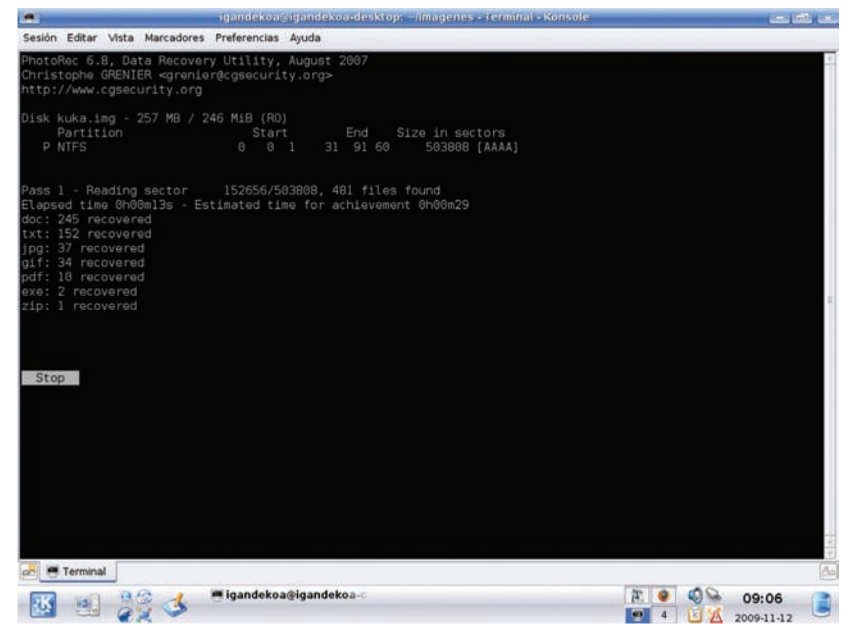


Figura 6. Photorec recuperando archivos

Si Photorec no está instalado en el sistema podemos obtenerlo mediante el gestor de paquetes o desde la consola:

```
local@local-desktop:~$ sudo apt-get
install testdisk
```

Photorec puede ser ejecutado contra el soporte de datos (*/dev/sdd*) o contra la imagen. Lo último resulta preferible no solo por razones de precaución sino también de rendimiento. Aunque se trata de un soporte antiguo, su imagen está grabada sobre un disco duro PATA con altas velocidades de transferencia y una capacidad total de 320 GB. Por lo tanto:

```
local@local-desktop:~/mnt/hdb1/
imagenes_forenses$ photorec imagen_
disco1.dd
```

Dentro de *Photorec* el usuario encontrará un número de opciones para experimentar. Con la configuración por defecto pueden realizarse operaciones de rescate que no tienen nada que envidiar al más sofisticado programa de recuperación de archivos en Windows. Los resultados, a diferencia de *Foremost* o *Scalpel*, se guardan sin clasificar dentro de una serie de directorios numerada 'recup_dir.1' a 'recup_dir.n', creándose un nuevo directorio cada vez que se extraen 500 archivos.

Tipos de archivos

El examen de la evidencia se lleva a cabo mediante las herramientas de trabajo del sistema: OpenOffice.org, visores de gráficos, editores de texto, etc. Si hay archivos intactos de Outlook Express, se pueden importar desde Evolution, con lo que se tendrá acceso al correo electrónico del usuario. La actividad de Internet se puede reconstruir mediante *Pasco* a partir de los archivos de registro de Internet Explorer (*http://www.foundstone.com/us/resources/proddesc/pasco.htm*).

Resulta sorprendente la cantidad de archivos que se puede extraer de un disco duro antiguo con Photorec: además de multimedia y gráficos para una amplia gama de cámaras fotográficas, documentos Office, páginas HTML, ejecutables y librerías del sistema, bases de datos, archivos comprimidos en ZIP, RAR, TGZ, CAB, documentos Acrobat PDF, archivos de texto y muchos otros. Las nuevas versiones de este programa incluso pueden asignar a los archivos JPEG la fecha y hora incluidas en las respectivas cabeceras EXIF.

Resultados

En casi todos los soportes se hallaron datos personales. En ocasiones el disco ni siquiera había sido borrado, lo cual revela que todos esos avisos de eBay del tipo "el disco se vende comprobado y formateado" no son más que cháchara de bazar. Los vendedores no se toman la molestia de conectar sus unidades a un interfaz IDE para reformatearlas, y no digamos llevar a cabo un borrado seguro. En tres o cuatro casos -de una veintena de discos analizados- el sistema operativo estaba completo. Incluso podía arrancar desde una máquina virtual.

Aparte de los archivos de sistema, librerías y software instalado por el usuario, los objetos de mayor interés en un disco duro de ocasión son documentos de texto -no solo Office, sino también otros formatos: Word Perfect, AmiPro o MS Works- y gráficos JPG, descargados de Internet o procedentes de cámaras digitales. Hay currículums vitae, trabajos escolares, declaraciones de la renta y toda una colección de objetos de carácter confidencial y privado.

En los discos procedentes de la universidad había documentos sobre un proyecto de automatización industrial, trabajos de cientos de páginas, correspondencia de los docentes, memorias y solicitudes de ayudas dirigidas a la Administración y las grandes empresas. Otro de los discos duros, esta vez formateado, contenía gran cantidad de nombres, direcciones y números de teléfono, perfectamente recuperables de la imagen mediante un editor hexadecimal o scripts de Perl.

En uno de los discos duros aparecieron los archivos de una cofradía de Semana Santa, con un listado de los miembros, otro de gente retrasada en el pago de las cuotas y actas de la junta que revelaban la existencia de conflictos entre los cofrades. El secretario de la junta trabajaba como agente comercial de varias empresas, cuya documentación, listas de precios, correspondencia comercial y demás también aparecía pulcramente ordenada en carpetas.

De las tarjetas SD mejor no hablar, pues de ellas resulta todavía más fácil extraer material que de los discos duros. Debido al carácter personal de estos datos, aquí el usuario sí que debería mostrarse bastante más precavido. A no ser que sepa borrarlas de modo seguro, en ningún caso tendría que regalarlas ni venderlas en el mercado de ocasión. Después de haber visto lo que se puede sacar de ellas, lo único que puedo decir con toda honestidad es que a mí jamás se me ocurriría hacerlo. En una de dichas tarjetas, que al parecer también había sido utilizada como dispositivo de almacenamiento en un PDA, había hojas Excel con datos técnicos correspondientes al cuarto de máquinas de un hotel.

Máquinas virtuales

Tan solo a modo de curiosidad técnica que constituye una prueba más de la potencia y versatilidad de Linux aplicadas a la investigación forense, algunos de estos discos duros se hallan en tan buenas condiciones que resulta posible arrancarlos mediante software de virtualización. Hice pruebas con el más antiguo de todos, uno de aquellos legendarios Conner que en las postrimerías del milenio eran paradigma de excelencia tec-

```
igandekoa@igandekoa-desktop: ~/imagenes - Terminal - Konsole
Sesión  Editar  Vista  Marcadores  Preferencias  Ayuda
PhotoRec 6.8, Data Recovery Utility, August 2007
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk kuka.img - 257 MB / 246 MiB (RO)
     Partition                  Start        End    Size in sectors
   P NTFS                     0   0  1    31  91 60      503808 [AAAA]  Stop


Pass 1 - Reading sector      208368/503808, 487 files found
Elapsed time 0h00m28s - Estimated time for achievement 0h00m39
doc: 245 recovered
txt: 152 recovered
jpg: 43 recovered
gif: 34 recovered
pdf: 10 recovered
exe: 2 recovered
zip: 1 recovered
Terminal
```

Figura 7. Photorec recuperando archivos de diversos tipos letra gruesa

nológica, con un tamaño de 210 MB -sí, han leído bien: megas, no gigas-. Esta maravilla resultó tener intacto el sistema operativo (DOS 3.0 + Windows 3.1). Mediante las utilidades en línea de comando de *Qemu*, conocido software de virtualización de dominio público, convertí la imagen adquirida con 'dd' en un archivo .vmdk, que es el formato utilizado por VMware:

```
local@local-desktop:~/mnt/hdb1/
imagenes_forenses$ qemu-img convert
-f raw imagen.dd -O vmdk imagen.vmdk
```

Acto seguido creé con VMware Workstation una máquina virtual DOS. El resto fue reinstalar los archivos de sistema DOS desde un viejo diskette, realizar ajustes menores en los archivos AUTOEXEC.BAT y CONFIG.SYS y cambiar la configuración del mouse, que en aquellos tiempos se solía conectar al ordenador mediante puerto serie (COM1) a través de un interfaz serie RS232 y no mediante el PS2 actual. Al cabo de pocos minutos pude contemplar como regresaba a la vida un ordenador del Pleistoceno, con su flamante entorno de ventanas Windows 3.1, tal y como su antiguo propietario lo había visto antes de llevarlo al desguace.

Cómo acabar con los datos antiguos de una vez por todas

¿Constituye el borrado seguro una solución? Existen programas capaces de eliminar datos sin posibilidad de recuperación sobreescribiéndolos varias veces con información aleatoria. Sin embargo, ¿cómo estar seguro de que no quedan copias del archivo en otras ubicaciones del disco duro, carpetas temporales, la partición de intercambio o el archivo de paginación? Y tratándose de Windows 2000, XP o Vista sobre un sistema de archivos NTFS, los metadatos permanecerán en la MFT. Aunque no pueda leerse el contenido, se sabrá que el archivo estuvo allí. Una manera práctica de higienizar un disco duro, laminando por completo todas sus estructuras de datos y dejándolo como recién salido de fábrica, nos la proporcionan una vez más las herramientas de código libre. 'dd' permite sobreescribir todo el soporte con ceros:

```
local@local-desktop:~$ sudo dd if=/
dev/zero of=/dev/hdd,
```

o si se prefiere, con caracteres aleatorios:

```
local@local-desktop:~$ sudo dd if=/
dev/urandom of=/dev/hdd
```

Tampoco esto es seguro al cien por cien, ya que existen métodos de procesamiento de señales magnéticas que permiten recuperar archivos de datos incluso después de haber sido sobreescritos varias veces. Pero en la práctica, dadas las dificultades técnicas y la densidad de datos cada vez mayor de los discos modernos, la recuperación resulta imposible o al menos prohibitiva debido a la inversión de trabajo y costes que implicaría, incluso disponiendo de un equipo avanzado.

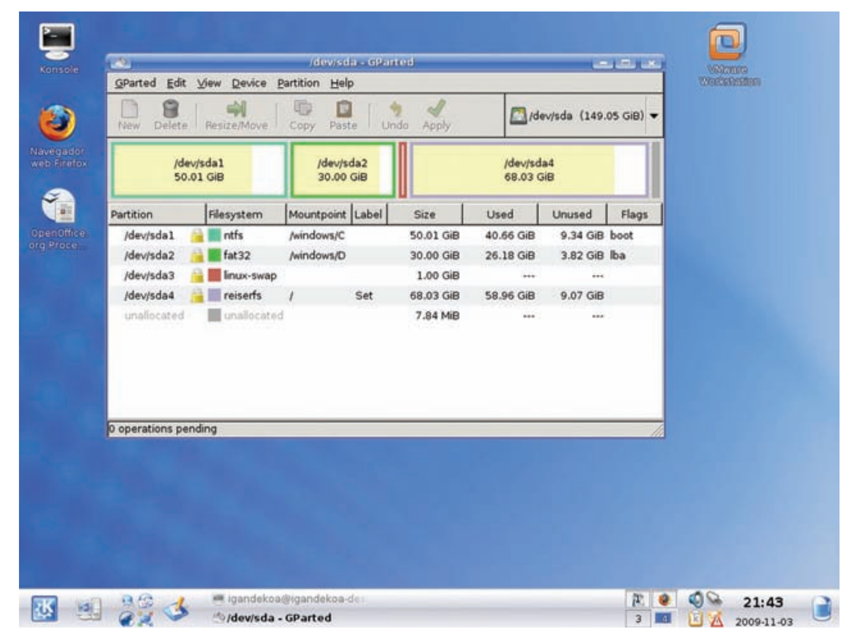


Figura 8. Visualizando particiones de un disco duro con gparted

A propósito: el usuario no debe ponerse a experimentar alegremente con 'dd' y privilegios de root en su estación de trabajo, a no ser que esté seguro de lo que hace.

El borrado seguro de un soporte de datos requiere tiempo, tanto más cuanto mayor sea la capacidad del soporte, hasta varias horas para los más modernos discos duros PATA o SATA con capacidades en el orden de los centenares de gigabytes. Posiblemente sea esta una de las razones que explica la negligencia de los vendedores en eBay.

Conclusiones

Hurgar en datos ajenos no resulta agradable. A más de un lector el enfoque de las páginas anteriores le parecerá algo frívolo, pero creo que es mejor poner de manifiesto, sin eufemismos ni perífrasis, los penosos niveles de seguridad existentes en cuanto a la eliminación de material informático usado. Los datos personales y otras informaciones sensibles están al alcance de cualquiera. Téngase en cuenta, además, que estos discos proceden en su mayor parte de usuarios particulares. Podemos imaginar lo que puede dar de sí el análisis de discos duros comprados en el ámbito empresarial, subastas o similares. Y no digamos cuando se pierden soportes pertenecientes a las administraciones públicas con datos personales de los ciudadanos, registros de un hospital o información sobre testigos protegidos.

En el magazine digital IEEE Security & Privacy donde Garfinkel y Shelat publicaron los resultados de su estudio, Garfinkel concluye: "Si fuera un espía interesado en conocer la vida económica de un país, gastaría un millón de dólares al año para comprar discos duros y analizarlos". Esto se escribió hace seis años y sigue siendo válido. Aparte del interés indudable de Linux y las herramientas de código libre para la investigación forense, está claro que debería haber mayor conciencia pública sobre el tema. Contribuir a ello es el principal objetivo de este artículo.

En la red

- *http://simson.net/clips/academic/2003.IEEE.DiskDriveForensics.pdf*
- *http://es.wikipedia.org/wiki/Disco_duro*
- *http://www.usenix.org/publications/library/proceedings/sec96/full_papers/gutmann/*

DNS: Domain Name System

Roberto Andradas Izquierdo

Con la explosión del número de hosts conectados a la red, que sucedió tras la creación de ARPAnet en 1970 y la implantación del protocolo TCP/IP, la gestión de los nombres de cada uno de ellos y sus direcciones en la red se volvió realmente costosa y poco escalable, lo que provocó el diseño de un nuevo servicio de red llamado DNS. Este nuevo servicio con el tiempo se ha convertido en uno de los pilares de lo que ahora conocemos como Internet.



linux@software.com.pl

En los inicios de ARPAnet, alrededor de 1970 cuando únicamente existía una pequeña cantidad de hosts conectados entre ellos, la gestión de los nombres de estos se hacía mediante un fichero de texto llamado HOSTS.TXT que contenía una entrada "dirección-nombre" que asociaba la dirección de un host en la red con un nombre fácil de recordar por cada línea. De este modo cada usuario de la red podría referirse a otro host utilizando su nombre ya que los sistemas de la época usaban dicho fichero para conocer la dirección asociada a ese nombre.

Cada host tenía una copia de este fichero que era descargada de un servicio de información de red que mantenía la copia maestra. De este modo cada vez que un nuevo host aparecía en la red su dueño comunicaba a dicho servicio su presencia y por lo tanto era añadido al fichero HOSTS.TXT maestro, los hosts de la red descargaban este fichero con cierta frecuencia para conocer los nuevos miembros de la red. Este sistema centralizado funcionaba pero tenía problemas graves como su poca escalabilidad, el tráfico generado en la red para actualizar el fichero en todos los hosts crecía debido al creciente tamaño del mismo y al número de hosts que poblaban ARPAnet que cada vez era mayor, la probabilidad de un cambio en el fichero HOSTS.TXT maestro aumentaba según crecía la red por lo que era imposible mantener un fichero HOSTS.TXT actualizado en cada host de la red pues a los pocos minutos de una descarga la copia maestra podría haber cambiado de nuevo siendo necesaria otra. Este servicio centralizado que distribuía dicho fichero era llamado NIC, no tenía autoridad sobre los nombres añadidos a la copia maestra por lo que en ocasiones había múltiples entradas para el mismo nombre. Todos estos problemas realmente podían haber acabado con ARPAnet pues impedían su crecimiento, su estabilidad y su eficiencia.

La eliminación del cuello de botella que suponía tener un sistema centralizado pasaba por crear un sistema descentralizado de administración de nombres. Además, dicho sistema debería permitir la administración local de nombres de forma que cada entidad pudiera gestionar sus propios nombres y que estos fueran conocidos desde cual-

quier punto de la red. Organizar los nombres de forma jerárquica garantizaría la unicidad de los nombres. A este nuevo sistema de nombrado de máquinas se le llamo "Domain Name System" o Sistema de Nombres de Dominio.

Representación de DNS

La mejor forma de entender y conocer DNS es encontrando una buena forma de representarlo, para ello vamos a utilizar un árbol como el de la Figura 1. Dicha estructura es muy adecuada para representar datos que están organizados de forma jerarquizada. En dicha figura, a primera vista observamos un nodo raíz denominado con el carácter ".", algunos nodos que nos son familiares y otros que no lo son. El nodo raíz es aquel del cual cuelgan el resto, es bajo el cual están todos los demás, representa nuestro origen de nombres de dominio.

A continuación observamos los nodos "com", "org", "edu" y "gov" que suelen escribirse con un "." delante como haremos a partir de ahora. Representan los llamados dominios de primer nivel o TLDs (Top Level Domains) y originalmente fueron creados para asociar cada nombre de dominio a una temática, todos los nombres de dominio están bajo uno de estos nodos:

- *.com*: hosts relacionados con actividades comerciales.
- *.gov*: hosts relacionados con actividades gubernamentales.
- *.edu*: hosts relacionados con actividades educativas.
- *.es*: hosts relacionados con España.
- *.org*: todos los demás, es decir aquellos que no tienen una temática que pueda asemejarse al resto de dominios de primer nivel. Por ejemplo organizaciones no lucrativas.

Actualmente existen 274 dominios de primer nivel.

Dentro de los dominios de primer nivel existen dos tipos, gTLDs (Global Top Level Domains) y ccTLDs (Country Coded Top Level Domains). Los primeros son aquellos que no están asociados a un país como por ejemplo ".com" y los segundos son los asociados a un país como es el caso de ".es" para España.

Dentro de cada nivel del árbol no pueden existir dos nodos con el mismo nombre, esto resuelve el problema de la duplicidad de nombres. Es decir, en el nivel 1 no puede haber dos nodos llamados "com".

Aquellos nodos con hijos los denominaremos subdominios, de modo que el nodo "bar" que cuelga de "com" es un subdominio, dicho subdominio recibirá el nombre "bar.com". Escribirlo así garantiza la identificación unívoca. Esta gestión de los nombres permite delegar a las organizaciones que corresponda su parte del árbol, es decir, la organización bar es la encargada de gestionar "foo" y "susan". Además, observamos que del nodo "bar" cuelgan nodos hoja, aquellos sin hijos, y otros subdominios como por ejemplo "foo" que como imagináis se escribirá "foo.bar.com". Los primeros representan hosts físicos de la red y los segundos simplemente son subdominios que podrían ser delegados a departamentos de la organización bar.

Como veis se puede delegar cada parte del árbol (subdominios) a la organización, departamento o persona que deseemos. De este modo evitamos la gestión centralizada de los nombres y evitamos el cuello de botella que supuso la utilización del fichero HOSTS.TXT.

Se observa de nuevo en la Figura 1 que además existen lo que denominamos zonas. No son exactamente lo mismo que los subdominios y es muy importante aclarar y entender las diferencias para posteriormente realizar una correcta configuración del software que implementa el servicio de DNS.

Cuando hablamos de subdominio nos estamos refiriendo a todo el árbol que cuelga por debajo de un nodo. Es decir, el subdominio "bar.com" es el árbol formado por los nodos "susan", "foo" y todo lo que cuelga de foo y a la vez de sus hijos hasta las hojas situadas en el nivel más bajo del árbol.

En cambio, una zona contiene la información que cuelga directamente de uno de los nodos, por ejemplo la zona "bar" del subdominio "bar.com" contiene únicamente la información referente a "foo" y a "susan". "susan" es un host luego será una dirección y "foo" es otro subdominio lo cual será un puntero a la organización/departamento al cual está delegado dicho subdominio.

¿Cómo funciona?

La finalidad de este sistema es conseguir la dirección de un host de la red dado un nombre de dominio. Esta dirección viene expresada por un valor llamado dirección IP que es una cadena con 4 cifras separadas por el carácter "." como por ejemplo: 192.168.0.45. A esta tarea se le llama resolución de nombre de dominio.

Esta conversión es realizada por un software que implementa la lógica descrita anteriormente. Habitualmente este software recibe el nombre de "servidor dns". Es el encargado de gestionar un subdominio que ha sido delegado, es decir, cuando a la empresa “bar” se le delega el subdominio "bar.com", esta utiliza este tipo de software para realizar dicha gestión. Sin una delegación explícita hecha por “.com” dicho servidor jamás recibirá peticiones del resto de Internet para resolver nombres del subdominio “bar.com”, por lo que es importante asegurarse que el subdominio está correctamente delegado.

La mejor forma de entender cómo funciona la resolución de un nombre de dominio es con un ejemplo básico.

Supongamos el nombre de dominio "fiona.foo.bar.com", dicho nombre pertenece a un host cuya propiedad es de Fiona. Además, Fiona trabaja en el departamento "foo" de la empresa "bar". En este host se encuentra alojada una página web personal donde Fiona publica textos de interés para otras personas de Internet. Imaginemos además que estás interesado en visitar la página de Fiona. Obviamente dada la inmensidad de Internet y la imposibilidad de aprenderse una combinación de números como la anterior, para cada una de las páginas web que visitamos a diario, lo único que recordamos es que la dirección de la web es "http://fiona.foo.bar.com", la cual introducimos en nuestro navegador. A partir de este momento, antes de empezar a recibir datos de dicha página, se desarrolla una cadena de acontecimientos llamada resolución del nombre que en apenas milisegundos permitirán, a nuestro ordenador personal, saber la dirección del host que tiene dicha página web y utilizarla para pedir los datos que queremos al lugar correcto.

Muchos habréis observado cómo vuestro ordenador personal tiene configurado la dirección IP de un servidor DNS. ¿Por qué no se usa un nombre? Porque no podemos usar un nombre para intentar encontrar el servidor que resuelve nombres, parece obvio pero es probablemente uno de los pocos casos en los que la no utilización de nombres de dominio es razonable. Este servidor DNS, al cual llamaremos A, es a quién preguntamos algo así como: ¿cuál es la dirección del nombre de dominio "fiona.foo.bar.com"?

El servidor DNS A actúa en modo forwarder, como la mayoría de los servidores DNS proporcionados por nuestro proveedor de acceso a Internet, esto significa que permite peticiones recursivas. O lo que es lo mismo, recoge

la petición del usuario y se encarga de obtener una respuesta definitiva a su petición, para ello recorre el nombre de dominio de derecha a izquierda. En primer lugar, a pesar de que habitualmente se omite en los nombres de dominio como hemos hecho nosotros, tenemos que observar que el primer carácter de un nombre de dominio empezando por la derecha es el ".", es decir, en realidad el nombre de dominio por el que preguntamos es "fiona.foo.bar.com.". Una vez aclarado esto, parece obvio que al recorrer de derecha a izquierda el nombre, el servidor DNS A encuentra el carácter ".". Esto le indica que debe preguntar por dicho nombre a los servidores raíces.

Por lo tanto, DNS A pregunta a los servidores raíz, es decir a los hosts que gestionan el dominio ".", por la dirección IP del nombre que nos ocupa. Ellos únicamente conocen información de la zona "." por lo que lo más que podrán decir a DNS A es que pregunte al servidor DNS en el cual está delegado el subdominio "com". A continuación, como paso 4, el servidor DNS A pregunta al servidor encargado de "com" por la dirección "fiona.foo.bar", al igual que en el paso anterior, este servidor únicamente tiene información de la zona "com" por lo que como mucho sabe quién es el servidor dónde está delegado el subdominio "bar", lo cual es enviado a DNS A. Seguimos preguntando al servidor dónde se delegó el subdominio "bar" por la dirección "fiona.foo", dicho servidor nos dirá que preguntemos a quién se encarga de "foo" proporcionándonos la dirección IP de dicho servidor DNS como se hizo en los pasos anteriores. Finalmente, al preguntar al servidor encargado del subdominio "foo", el cual contiene información de dicha zona, obtendremos la IP de "fiona" que será enviada al forwarder y este se la enviará al usuario para ser utilizada por el navegador. Además, este dato será almacenado en la cache de DNS A por si el usuario vuelve a preguntar una segunda vez a DNS A por el mismo nombre. En este caso se devolverá el valor almacenado en su cache. Lugar donde el dato es almacenado durante un tiempo definido en el servidor encargado de la zona "foo".

Los servidores donde se han delegado subdominios habitualmente no son forwarders y solo permiten peticiones como las que ha hecho DNS A, es decir, peticiones de tipo iterativo. Pueden existir varios servidores encargados de un subdominio, en este caso uno de ellos será el maestro y otros los esclavos. Es el servidor encargado del subdominio inmediatamente superior en el árbol el que aplica algoritmos habitualmente de tipo Round Robin a

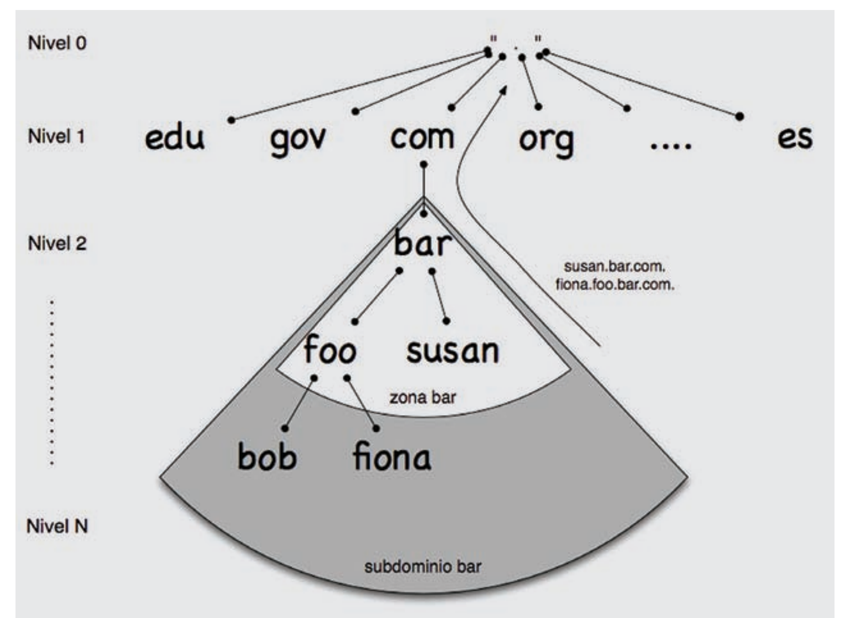


Figura 1. Ejemplo de la estructura del árbol de nombres de dominio

la hora de contestar a quién se debe hacer la siguiente petición, de modo que las preguntas se reparten entre todos los servidores, maestro y esclavos, encargados de un subdominio. Por ejemplo, si el subdominio “foo.bar.com” está gestionado por un servidor maestro y 3 esclavos, que replican la información del maestro, entonces el servidor encargado de “bar.com” aplicará el algoritmo Round Robin para dirigir a quién pregunta por algún nombre de “foo” hacia alguno de los servidores encargados de dicho subdominio.

Configuración de un servidor DNS

En primer lugar, es muy importante comprender lo explicado en la sección anterior pues sin este conocimiento la configuración de servidores DNS para la gestión de subdominios resultará fallida y aunque es probable que el servicio funcione estaremos cometiendo errores que en entornos donde manejamos cientos o miles de nombres provocarán una difícil y poco eficiente gestión del servicio con su consecuente pérdida de tiempo y dinero. Por otro lado, lo que se verá a continuación es un ejemplo de configuración, hay multitud de otras opciones que no caen dentro del foco de este artículo que podrían resultar de tu interés. Por último, antes de comenzar se debe tener en cuenta que las direcciones IP utilizadas en este artículo pertenecen a redes privadas para evitar conflictos o pruebas por parte del lector con direcciones IP públicas que podrían pertenecer a organizaciones reales.

Existe numeroso software para la creación de servidores DNS, en este artículo vamos a utilizar BIND9 en una supuesta plataforma GNU/Debian Lenny ya instalada, concretamente el paquete bind9 versión 1: 9.5.1.dfsg.P3-1.

Suponemos que vamos a registrar el nombre de dominio "bar.com" en Internet, para ello debemos proporcionar al menos una dirección IP del servidor al cual se debe hacer la delegación de dicho subdominio. Esta dirección será la del servidor que vamos a configurar a continuación. Debe ser una dirección IP accesible desde Internet. De lo contrario nadie en Internet podrá resolver nombres del subdominio "bar.com".

Una vez registrado el dominio nos ponemos manos a la obra en la configuración de nuestro servidor DNS.

```
# apt-get install bind9 bind9utils
# cd /etc/bind
```

Los servidores donde se delega la gestión de los subdominios se llaman servidores autorizados. Tanto servidores maestros como servidores esclavos serán considerados servidores autorizados y los clientes que les hagan peticiones serán incapaces de distinguir cual es el maestro y cual es el esclavo, siempre recibirán una respuesta autoritativa.

Nosotros, dado que vamos a configurar un servidor autorizado del subdominio "bar.com", vamos a dar de alta en el servidor bind9 dicha zona. Para lo cual editamos el fichero "named.conf.local" añadiendo la zona como se indica en el Listado 1.

Hemos añadido una zona llamada "bar.com", además hemos dicho que ese servidor

será el maestro y que el fichero con la información de la zona será "/etc/bind/db.bar.com". A este fichero, también se la llama mapa de zona. Además permitimos un servidor esclavo que tiene que tener la dirección IP indicada en "allow-transfer". Es muy importante definir quiénes podrán ser esclavos del maestro, esto evitará que personas con intenciones no muy buenas puedan obtener una copia de nuestro mapa de zona y por lo tanto una pista estupenda del diseño de nuestra red y los nombres de nuestras máquinas lo que suele dar una pista bastante buena del propósito de las mismas.

A continuación debemos editar el fichero como aparece en el Listado 2.

$TTL es el tiempo de vida que tendrá la información de la zona "bar.com" en las caches de los servidores DNS que la almacenen. Este valor puede ocasionar una pequeña falta de sincronización si cambiamos la dirección de un host. Afortunadamente, controlamos ese valor y podemos modificarlo según consideremos. Se desaconseja poner valores inferiores a 3h para evitar la congestión de los servidores autorizados, esto podría protegernos de un ataque de denegación de servicio en el cual una cantidad enorme de máquinas pregunten por un nombre del subdominio que gestionamos ya que las respuestas a dicha cantidad de preguntas serían dadas por caches de otros servidores en vez de nosotros. Tampoco es bueno poner un $TTL demasiado grande para evitar inconsistencias entre el dato real y el dato almacenado en las caches del resto de servidores DNS de Internet cuando se realiza un cambio de IP para un nombre.

SOA, es un registro especial donde decimos en primer lugar que el servidor DNS primario de la zona es "dns1.bar.com" y que el correo electrónico de la persona que lo administra o persona de contacto es dnsmaster@bar.com. Debemos sustituir el primer "." por @ para darnos cuenta de cual es la dirección de correo que representa. A continuación en el registro SOA encontramos una serie de parámetros:

- *Serial*: es la versión del fichero, cada vez que realizamos una modificación debemos aumentar dicho número. Tiene el formato YYYYMMDDCC donde YYYY es el año, MM el mes, DD el día y CC un contador de modificaciones hechas a lo largo del día. Es obligatorio y muy importante como veremos más adelante actualizar el valor cada vez que modificamos el fichero.
- *Refresh*: es el valor que especifica cada cuanto tiempo los servidores esclavos intentarán sincronizarse con el servidor maestro.
- *Retry*: si durante un intento de sincronización no se pudo conectar con el servidor maestro en este campo indicamos cual será el tiempo con el que repetiremos desde el esclavo la sincronización.
- *Expire*: si durante el tiempo especificado en este campo no hemos conseguido sincronizar con el servidor maestro, el servidor esclavo dará de baja la zona automáticamente y no la servirá.
- *Negative*: es utilizado en los servidores que actúan como forwarders, indica el tiempo que se almacena en cache un mensaje del tipo "nombre de dominio no existe".

"Default A record", es la entrada por defecto. Los subdominios también pueden tener una dirección IP. No tiene por qué pertenecer a ningún host, aunque se recomienda que sí pertenezca. Se puede deducir de aquí que un subdominio puede estar asociado a un host que podría contener por ejemplo otros servicios.

"NS records", en esta parte se indica cuales son los servidores DNS de la propia zona "bar.com". Como se observa, está él mismo y otro que por el nombre de dominio que le identifica se deduce que es un servidor esclavo de la zona. Además, es interesante fijarse cómo al final del nombre de dominio de ambos servidores de nombres se ha incluido el ".", esto sirve a bind9 para saber que son nombres absolutos, de otro modo pensaría que son nombres de la zona "bar.com" y podríamos sufrir algún que otro dolor de cabeza intentando encontrar por qué las cosas no funcionan.

"MX records", simplemente indican donde está el servidor de correo que tiene los buzones de los usuarios de bar.com. Es decir,

Figura 2. Resolución paso a paso del nombre de dominio fiona.foo.bar.com

Listado 1. Fichero /etc/bind/named.conf.local

```
   # vim named.conf.local
   zone "bar.com" {
      type master;
      file "/etc/bind/db.bar.com";
      allow-transfer { 192.168.99.101; };
   }
```

Listado 2. Fichero /etc/bind/db.bar.com

```
$TTL  3h
@  IN SOA   dns1.bar.com   dnsmaster.bar.com.  (
                                 2009091901  ; Serial
                                 1h          ; Refresh after 3 hours
                                 1h          ; Retry after 1 hour
                                 1w          ; Expire after 1 week
                                 1h )        ; Negative caching TTL of
1 hour

; Default A record
               IN A     192.168.41.15

; NS records
               IN NS    dns1.bar.com.
               IN NS    dns2.slavebar.org.

; MX records
               IN MX    10 mail

; Subdomains delegation
foo                     86400    IN NS    dns1.foo.bar.com.
                        86400    IN NS    dns2.slavefoo.org.
dns1.foo.bar.com.       86400    IN A  192.168.41.16
dns2.slavefoo.org.      86400    IN A  192.168.67.88

; Host addresses
localhost      IN A     127.0.0.1
dns1           IN A     192.168.41.16
susan          IN A     192.168.41.20

; Aliases
www            IN CNAME    susan
```

Listado 3. Fichero /etc/bind/named.conf.local

```
   zone "bar.com" {
      type slave;
      masters { 192.168.41.16; };
      file "/etc/bind/db.bar.com.slave";
   }
========
```

esta entrada en el mapa de la zona responde a la pregunta ¿a dónde debe llegar un email enviado a la dirección usuarioxxxx@bar.com?

"Subdomains delegation", es aquí donde realizamos la delegación de la zona foo.bar.com. En este caso, se indica cuales son los servidores DNS autorizados para esta zona y sus IPs. Es necesario apuntar sus IPs ya que nuestro servidor autorizado para bar.com no realiza peticiones recursivas y sólo sabe lo que pone en su mapa de zona por lo que no sabría resolver los nombres indicados en las entradas NS de esta sección.

"Host addresses", sección donde indicaremos las direcciones IPs de cada uno de los nombres de dominio que hemos usado en el mapa de zona. Vemos como los nombres situados en la columna de la izquierda no terminan en ".", esto es así porque son nombres de la zona "bar.com", serían las entradas correspondientes para peticiones de nombres de dominio del tipo "dns1.bar.com" o "susan.bar.com". Es típico incluir una para localhost, además incluiremos otra para dns1 y otra para el host "susan".

"Aliases", en esta sección incluimos nombres que son alias o equivalentes a los de la sección anterior. Esto evita que tengamos que mantener varias entradas para la misma dirección IP, teniendo que cambiar todas ellas en caso de que cambie dicha dirección. En este caso, existe un nombre de dominio "www.bar.com" que resulta tener la dirección del ya conocido host "susan".

Una vez hecho esto ya tenemos todo lo necesario para poner a funcionar nuestro servidor maestro. Para ello basta con utilizar el comando "rndc" como sigue:

```
# rndc reload bar.com
```

Es recomendable utilizar esta herramienta en vez del clásico "/etc/init.d/bind9 restart" porque un servidor con mapas de zona muy grandes reiniciar el servidor entero puede ocasionar un pequeño tiempo en el que el servicio no esté disponible con la consecuente mala imagen para el usuario. El comando "rndc" recarga únicamente la zona indicada.

Debemos indicar que es recomendable colocar los servidores maestros y esclavos en redes distintas para aumentar la disponibilidad de nuestro servicio de resolución de nombres. En caso de que una red tuviera problemas, el resto de servidores encargados del subdominio en cuestión podrían continuar operativos.

El servidor esclavo simplemente requerirá tener instalado bind9 y la zona de la cual es servidor esclavo añadida a su fichero "/etc/bind/named.conf.local" como se ve en el Listado 3. Se observa que ahora indicamos que el rol del servidor esclavo para esa zona es el de esclavo.

A estas alturas, se podría deducir que el mapa de zona de "foo.bar.com" tendrá un aspecto similar a cómo se observa en el Listado 4. Además, en el servidor maestro de "foo.bar.com" habrá que añadir dicha zona en el fichero "/etc/named.conf.local" como se hizo con la zona "bar.com". La única diferencia sustancial es la ausencia de la sección ";Subdomains delegation" puesto que no habrá ninguna delegación de subdominio.

En nuestro ejemplo, el servidor que se encarga del subdominio "foo.bar.com" es el mismo que el de "bar.com". No es necesario disponer de una máquina para cada subdominio, podemos añadir a un mismo servidor varios subdominios. A pesar de estar delegando en el mismo servidor subdominios diferentes y que podríamos caer en la tentación de eliminar la delegación de "foo.bar.com", añadir en el mapa de zona de "bar.com" una entrada como sigue:

```
; Host addresses
bob.foo      IN A   192.168.41.30
fiona.foo   IN A   192.168.41.31
```

Y eliminar la sección ";Subdomains delegation", lo cual funcionaría a la perfección a la hora de resolver nombres como "bob.foo.bar.com", no lo vamos a hacer porque estaríamos perdiendo la posibilidad de delegar en un futuro la zona foo.bar.com a otra organización. Incluso mover el mapa de zona a otra máquina podría convertirse en un problema pues habría que editar el fichero de la zona "bar.com" pudiendo romper alguna otra cosa.

En ocasiones, mapas de zona mal diseñados con fallos de este tipo han tenido que ser completamente reescritos pues ha sido imposible hacer las modificaciones necesarias debido a un cambio en la política de gestión de nombres de la organización que poseía la delegación. Así que, por favor, mantén los mapas de zona limpios, bien organizados y delega los subdominios de tus zonas sin dudarlo, ¡aunque sea a ti mismo! Os sorprenderíais de cuantos servidores DNS están mal configurados debido al empeño de sus administradores por utilizar los mapas de dominio como si fueran ficheros HOSTS.TXT.

Listado 4. Fichero /etc/bind/db.foo.bar.com

```
$TTL   3h
@   IN SOA   dns1.foo.bar.com   dnsmaster.foo.bar.com.   (
                                  2009091901   ; Serial
                                  1h           ; Refresh after 3 hours
                                  1h           ; Retry after 1 hour
                                  1w           ; Expire after 1 week
                                  1h )         ; Negative caching TTL of
1 hour
; Default A record
               IN A     192.168.93.10

; NS records
               IN NS    dns1.foo.bar.com.
               IN NS    dns2.slavefoo.org.
; MX records
               IN MX    10 mail
; Subdomains delegation
; Host addresses
localhost      IN A     127.0.0.1
dns1           IN A     192.168.41.16
bob            IN A     192.168.93.11
fiona         IN A     192.168.93.12
; Aliases
ftp            IN CNAME    bob
========
```

Conclusiones

Existe una organización destinada a gestionar los servidores raiz llamada ICANN, en su página web puedes encontrar todo tipo de material sobre gestión de dominios, compra, buenas políticas y por supuesto información de primera mano de la gestión de la zona ".", donde se encuentra la información necesaria para los TLDs.

Afortunadamente, debido al propio sistema de nombres de dominio las malas prácticas en la gestión del mismo no afectan a todo el sistema sino solo a los propios responsables de cada subdominio. Lo cual podría verse como otra buena cualidad del diseño distribuido de DNS. A su vez, esto nos está diciendo que somos responsables de nuestro propio manejo de nombres y por lo tanto responsables de la imagen de nuestra organización en Internet. Sin un servicio como DNS funcionando correctamente, simplemente, tus máquinas no son accesibles en Internet por un motivo muy sencillo, nadie va a aprenderse tus direcciones de memoria. De hecho, con toda probabilidad, muchos de tus propios servicios dejarían de funcionar.

Software como Bind9 permite multitud de configuraciones, temas como el balanceo de carga utilizando servidores DNS, resolución de nombres en función de la zona geográfica del usuario, zonas privadas y públicas y mucho más puede ser de tu interés por lo que no dudes en consultar bibliografía al respecto.



Sobre el autor

Roberto Andradas Izquierdo es Ingeniero Técnico en Informática de Sistemas. Nacido en Madrid, actualmente está estudiando Ingeniería en Informática además de trabajar como administrador de sistemas en el grupo de investigación de ingeniería del software libre GsyC/LibreSoft (*http://www.libresoft.es*) en la Universidad Rey Juan Carlos de Madrid. Participa en proyectos de proyección internacional a nivel europeo como OSOR.eu y a nivel nacional como morfeo-project.org. En esta revista es colaborador y sus intereses son principalmente el diseño de arquitecturas de sistemas basadas en servicios y procedimientos adecuados para el mantenimiento de dichas plataformas. Su sitio web personal es *http://www.randradas.org.* Podéis contactar con él a través de *randradas@gmail.com*.

389 Directory Server:
Alternativa libre al Active Directory de Microsoft

Alfonso Vera Rubio

En el contexto de las redes de ordenadores, se denomina directorio a una base de datos especializada que almacena información sobre los recursos, u "objetos", presentes en la red (tales como usuarios, ordenadores, impresoras, etc.) y que pone dicha información a disposición de los usuarios de la red.



linux@software.com.pl

Active Directory es el término utilizado por Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de computadores. Utiliza distintos protocolos principalmente LDAP, DNS, DHCP y kerberos. Su estructura jerárquica permite mantener una serie de objetos relacionados con componentes de una red, como usuarios, grupos de usuarios, permisos y asignación de recursos y políticas de acceso.

La pareja 389-Directory Server y Samba junto con DHCP, DNS nos permitirán ofrecer a los clientes Windows de nuestra red prácticamente la funcionalidad que obtendrían con un controlador de dominio Windows 2003.

Historia

389 Directory Server es la nueva encarnación de lo que fue el Fedora Directory Server, que en mayo de 2009 cambia su nombre a 389 para que el nombre del proyecto sea neutral con respecto al proveedor, y así facilitar su ejecución en otros Sistemas Operativos o sabores de Linux, pese a su novedad este proyecto es de los más longevos dentro del software libre, comienza con el proyecto original de slapd en la Universidad de Michigan; en 1996 los desarrolladores son contratados por Netscape y el proyecto pasa a ser conocido como NDS (NetScape Directory Server). Después de la adquisición de Netscape, AOL vende parte de la propiedad a Sun, pero se queda con algunos derechos que luego vende a Red Hat en 2005, comenzando así la andadura de este proyecto hasta nuestros días.

Características especiales:

- MMR (Replicación multimaster): la opción de escribir en dos o más maestros al mismo tiempo, con resolución automática de conflictos, proporciona solución a uno de los puntos críticos de OpenLdap, el no tener alta disponibilidad en las escrituras, (aunque creo que lo han solucionado en la última versión).
- Sincronización con Microsoft Windows: usuarios, grupos y contraseñas pueden ser sincronizados con los controladores de dominio Windows 2003/2000 y NT4.
- Mecanismos de Control de Accceso (ACLs). Ahora las ACLs se encuentran dentro de los propios datos,

frente a la gestión de ACLs del lado del servidor de OpenLdap.

- Disponibilidad 24x7: la administración y configuración (backup, modificación de esquemas y control de acceso) se puede realizar online sin necesidad de parar el servicio.
- Consola gráfica para administrar el servidor, usuarios y grupos.

Consideraciones iniciales:

- Nuestro servidor tiene una IP estática: 192.168.1.200
- En el fichero */etc/hosts* se necesita definir un nombre FQDN para la máquina: *lobezno.colegio.int*
- El servicio NTP debe estar correctamente configurado.
- Partimos de un sistema con Fedora 11 con todas la actualizaciones instaladas y los repositorios oficiales perfectamente configurados.
- Para ejecutar la consola de administración necesita tener una máquina virtual de Java en su versión 1.6 o superior, puede ser OpenJDK, GJC no funciona. Ejecutando *java -version* desde una consola puede conocer la versión instalada en su sistema.
- Para levantar el servicio de administración debe tener instalado el servidor Apache (no importa si no lo tiene ahora, al instalar 389-ds lo instalará como dependencia) en modo worker, para cambiar el modo de ejecución del servidor, diríjase al fichero */etc/sysconfig/httpd* y descomente la línea *HTTPD=/usr/sbin/httpd.worker*. No se olvide de ejecutar *chkconfig httpd on* para que el servidor web arranque automáticamente.

Instalación: YUM es tu amigo

Los usuarios de Fedora estamos de enhorabuena ya que la herramienta de gestión de paquetes YUM [2] se ha convertido en la herramienta por excelencia en la gestión de paquetes basados en RPM, versión a versión se hace cada vez más eficiente y más cómoda de usar. Ejecutando desde una consola como root: *yum install 389-ds* se encargará de resolver todas las dependencias y dejarlo listo para su configuración.

389 Directory server se encuentra en continuo desarrollo, los muchachos de Fedora trabajan como punta de lanza para los lanzamientos corporativos de Red Hat Directory Server y Red Hat IPA Server, si queremos saber qué se está cocinando en los laboratorios de Fedora, o testear lo que vendrá en próximos lanzamientos de los productos corporativos podemos ejecutar: *yum install –enablerepo=updates-testing 389-ds*. Esta opción no es recomendable para entornos en producción o si sufre del corazón.

Una vez instalado, siéntese cómodamente, nos disponemos a realizar la configuración inicial antes de levantar el servicio de directorio por primera vez. ¡Quietos! Antes de correr el script de configuración para poder arrancar

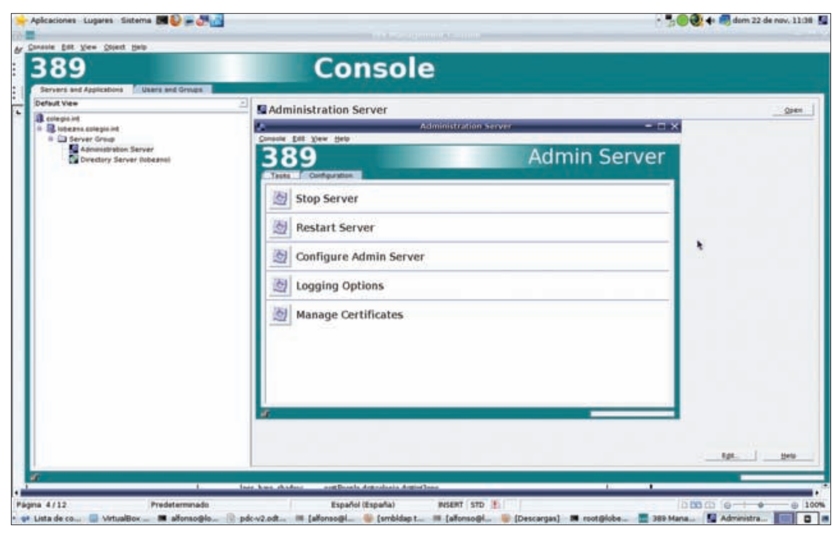


Figura 1. Server Admin

Listado 1. Afinando el Sistema Operativo

```
    echo "net.ipv4.tcp_keepalive_time = 300" >> /etc/sysctl.conf
    echo "fs.file-max = 64000" >> /etc/sysctl.conf
    echo "* soft nofile 8192" >> /etc/security/limits.conf
    echo "* hard nofile 8192" >> /etc/security/limits.conf
    echo "ulimit -n 8192" >> /etc/profile
    echo "64000" > /proc/sys/fs/file-max
    echo "net.ipv4.ip_local_port_range = 1024 65000" >> /etc/
sysctl.conf
```

Listado 2. Fichero /etc/ldap.conf

```
base dc=colegio,dc=int
ldap_version 3
binddn cn=Directory Manager
bindpw 12345
rootbinddn cn=admin,dc=colegio,dc=int
timelimit 120
bind_timelimit 120
idle_timelimit 3600
bind_policy soft
pam_filter objectclass=account
pam_login_attribute uid
nss_base_passwd  ou=People,dc=colegio,dc=int?one
nss_base_passwd  ou=Computers,dc=colegio,dc=int?one
nss_base_shadow  ou=People,dc=colegio,dc=int?one
nss_base_group   ou=Groups,dc=colegio,dc=int?one
nss_initgroups_ignoreusers root,ldap,named,avahi,haldaemon,dbus,radvd,
tomcat,radiusd,news,mailman,nscd,gdm,polkit
uri ldap://lobezno.colegio.int/
ssl no
pam_password md5
```

Listado 3. Fichero configuración Samba

```
[global]
#--------Parametros de red -------------------
---------------------------------------------
workgroup = COLEGIO
server string = Samba Server Version %v
netbios name = LOBEZNO
interfaces = lo eth0
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_
SNDBUF=8192
#--------------- Parametros de log -----------
---------------------------------------------
log file = /var/log/Samba/log.%m
max log size = 50
#---------- Controlador de Dominio------------
---------------------------------------------
security = user
domain master = yes
domain logons = yes
logon path =
os level = 65
preferred master = yes
encrypt passwords = yes
#----------------- Impresion------------------
---------------------------------------------
load printers = yes
cups options = raw
#--------------- Sistema de archivos----------
---------------------------------------------
profile acls = yes
nt acl support = yes
#------------------------- Ntp y Wins --------
---------------------------------------------
time server = yes
wins support = yes
#----------------Ldap ------------------------
---------------------------------------------
admin users = Administrator @"Domain Admins"
 passdb backend = ldapsam:ldap://lobezno.colegio.int

ldap suffix = dc=colegio,dc=int
ldap user suffix = ou=People
ldap group suffix = ou=Groups
ldap machine suffix = ou=Computers

ldap admin dn = cn=Directory Manager
ldap passwd sync = yes
ldap ssl = no

add machine script = /usr/sbin/smbldap-useradd -w %u
passwd program = /usr/sbin/smbldap-passwd %u
passwd chat = *New*password* %n\n
*Retype*new*password* %n\n *all*authentication*token
s*updated*
add user script = /usr/sbin/smbldap-useradd -m "%u"
ldap delete dn = Yes
delete user script = /usr/sbin/smbldap-userdel "%u"
add group script = /usr/sbin/smbldap-groupadd -p "%g"
delete group script = /usr/sbin/smbldap-groupdel "%g"
add user to group script = /usr/sbin/smbldap-groupmod
-m "%u" "%g"
delete user from group script = /usr/sbin/smbldap-
groupmod -x "%u" "%g"
set primary group script = /usr/sbin/smbldap-usermod
-g "%g" "%u"
#------------------ Directorios Compartidos---------
--------------------------------------------------]

[homes]
        comment = Home Directories
        browseable = no
        writable = yes

        valid users = %S

[printers]
        comment = All Printers
        path = /var/spool/samba
        browseable = no
        guest ok = no
        writable = no
        printable = yes

[netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        guest ok = yes
        writable = no
        share modes = no

[Profiles]
        path = /var/lib/samba/profiles
        browseable = yes
        guest ok = yes
        create mask = 0600
        directory mask = 0700

[public]
        comment = Directorio Publico
        path = /home/samba
        public = yes
        writable = yes
        printable = no
```

el servidor de directorio, necesitamos realizar algunos ajustes en nuestro sistema base para mejorar la eficiencia. No se preocupe, los chicos de Fedora piensan en todo, ejecute *dsktune* y realice las modificaciones sugeridas (ver Listado1).

Configuración básica

Desde una consola como root ejecute el script de configuración *setup-ds-admin.pl,* que le pedirá los datos básicos para poder arrancar el servicio. El script realizará una serie de preguntas que pasamos a detallar:

- En el tipo de instalación elija la opción por defecto: instalación típica,
- Inserte el nombre fqdn de nuestro servidor: lobezno.colegio.int,
- Dejamos por defecto el usuario y el grupo con el que correrá (nobody,nobody),
- No queremos unirnos a un directorio anterior (es nuestra primera instalación),
- Ahora es el turno de la contraseña del usuario admin (para loguearnos en la consola),
- El dominio: colegio.int,
- El puerto por defecto de ldap (389), el nombre del servidor y la raíz del directorio,
- Ahora el usuario "Directory Manager" para realizar operaciones con el directorio,
- Por último el puerto donde levantaremos la interfaz web del servidor de administración.

¡Et Voila! Eso es todo. Después de configurar el directorio el mismo script se ocupa de arrancarlo, para finalizar verifique la instalación usando ldapsearch, la utilidad ldapsearch se encuentra en el paquete openlap-clients que debemos instalar (si no está ya instalado). Ejecute después:

```
[root@lobezno ~]# yum install
openldap-clients
[root@lobezno lib]# ldapsearch
-x -h localhost  -s base -b ""
"objectclass=*"
```

Debemos asegurarnos de que los servicios arrancan en nivel 3, ejecute en una consola como usuario root *chkconfig dirsrv-admin on* y *chkconfig dirsrv on.*

Usamos la consola de administración (Figura 1, 2 y 3).

Integración con Samba

El directorio está instalado, hemos comprobado que funciona y hemos dado un vistazo a la consola de administración, llegados a este punto, intentaremos construir un PDC (Primary Domain Controler) para nuestra red con Samba usando como backend 389 directory Server.

El primer paso como siempre es la instalación de los paquetes necesarios: el servidor de Samba, el cliente de Samba, los ficheros comunes y el paquete de las herramientas de migración:

```
root@lobezno ~]# yum install
migrationtools Samba Samba-common
Samba-client smbldap-tools
```

Un esquema de LDAP define y controla los tipos de datos que pueden ser almacenados para entradas individuales en el directorio, nuestro directorio debe conocer el esquema de Samba para poder almacenar las entradas necesarias. La forma en que 389-ds carga los esquemas es algo distinta de como funciona en OpenLdap estándar, por lo que ahora vamos a repasar la carga de los esquemas de ldap en 389 directory server.

- El formato de los esquemas es siempre ldif, y se cargarán en el arranque del servicio,

Listado 4. Directorios Samba

```
mkdir -p /var/lib/samba
mkdir /var/lib/samba/{netlogon,profiles}
chown root:root -R /var/lib/samba
chmod 0755 /var/lib/samba/netlogon
chmod 1755 /var/lib/samba/profiles
```

Listado 5. Configuración de smbldap-tools

```
SID="S-1-5-21-XXXXXXXXX-XXXXXXXXX-XXXXXXXXX"
SambaDomain="COLEGIO"
slaveLDAP="127.0.0.1"
slavePort="389"
masterLDAP="127.0.0.1"
masterPort="389"
ldapTLS="0"
ldapSSL="0"
verify="none"
suffix="dc=colegio,dc=int"
usersdn="ou=People,${suffix}"
computersdn="ou=Computers,${suffix}"
groupsdn="ou=Groups,${suffix}"
SambaUnixIdPooldn="SambaDomainName=${SambaDomain},${suffix}"
scope="sub"
hash_encrypt="SSHA"
crypt_salt_format="%s"

userLoginShell="/bin/bash"
userHome="/home/%U"
userHomeDirectoryMode="700"
userGecos="System User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"

userSmbHome="\\LOBEZNO\%U"
userProfile="\\LOBEZNO\profiles\%U"
userHomeDrive="H:"
mailDomain="colegio.int"

with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
```

- Los esquemas se encuentran en: */opt/fedora-ds/slapd-<server>/config/schema,*
- Los esquemas se cargan de manera secuencial siendo siempre el último 99user.ldif.

En este documento vamos a llamar al esquema de Samba 61Samba.ldif, necesitamos convertir el esquema de Samba típico de ldap en uno que entienda nuestro directorio, para eso desde Fedora nos proporcionan un script para la conversión que podemos bajar desde aquí [4].

```
[root@lobezno ~]# perl ol-schema-
migrate.pl -b /usr/share/doc/
Samba-*/LDAP/Samba.schema > /
etc/dirsrv/slapd-lobezno/schema/
61Samba.ldif
```

Ahora reinicie el servicio para ver que carga el esquema correctamente.

PAM-LDAP

Vamos a integrar nuestro servidor Linux con el directorio mediante pam_ldap para que éste sea capaz de ver los usuarios y grupos necesarios. Esto se debe a que cuando pasemos a configurar el servidor Samba, los usuarios Samba tienen que tener su correspondiente usuario Linux y por lo tanto si el servidor Linux no es capaz de ver los usuarios como Linux que es, difícilmente podrá hacer que Samba los sirva. Ejecute *rpm -qa | grep -i nss_ldap* para verificar que tiene instalado el paquete nss_ldap, en el extraño caso de que no estuviera instalado en su sistema (viene con la instalación base), recurra a yum: *yum install nss_ldap*.

Los chicos de Fedora nos traen una estupenda herramienta para activar el uso de un directorio para obtener los usuarios y grupos, con lo que ya tenemos la mayor parte del trabajo hecho, si no se siente cómodo con la línea de comandos puede elegir usar authconfig-gtk, en modo gráfico, o ejecutar de un tirón desde la línea de comandos:

```
authconfig
--enableldap
--enableldapauth
--disablenis
--enablecache
--ldapserver=lobezno.colegio.int
--ldapbasedn=dc=colegio,dc=int
--updateall
```

Tanto si ejecuta authconfig desde la línea de comandos como si lo hace desde su escritorio favorito, las indicaciones son las mismas: habilitamos la obtención de información sobre los usuarios del directorio, permitimos autenticar usuarios mediante ldap, deshabilitamos nis, habilitamos la cache (demonio ncsd) y le pasamos el directorio y la base para realizar las búsquedas.

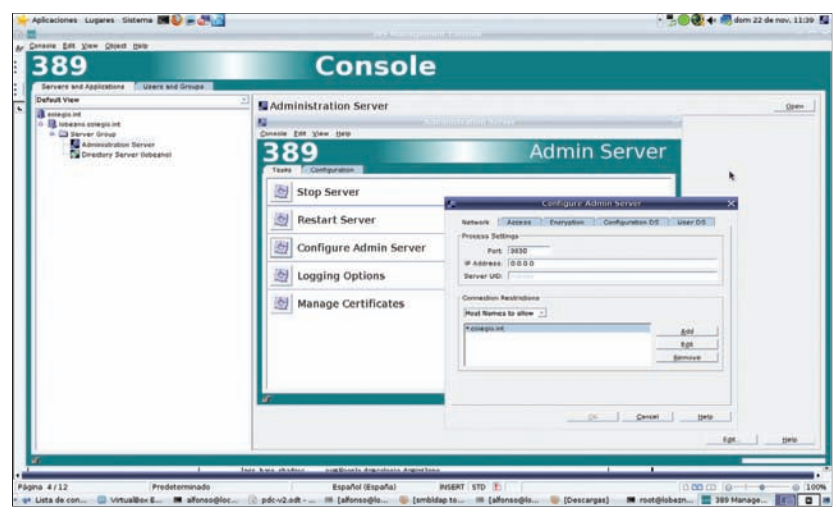


Figura 2. Configuración del servidor

No podemos fiarnos demasiado de las herramientas automáticas por lo que repasaremos el contenido de los ficheros modificados al ejecutar authconfig.

El archivo */etc/nsswitch.conf* determina en qué orden el sistema busca información sobre los host, usuarios grupos, redes etc.,

Listado 6. Salida del comando smbldap-populate

```
[root@lobezno ~]# service dirsrv start
Starting dirsrv:
    lobezno...                                        [  OK  ]
[root@lobezno ~]# service smb start
Iniciando servicios SMB:                              [  OK  ]
[root@lobezno ~]# smbldap-populate -a Administrator
Populating LDAP directory for domain COLEGIO (S-1-5-21-XXXXXXXXXX-
XXXXXXXXXX-XXXXXXXXXX)
(using builtin directory structure)
adding new entry: dc=colegio,dc=int
adding new entry: ou=People,dc=colegio,dc=int
adding new entry: ou=Group,dc=colegio,dc=int
adding new entry: ou=Computers,dc=colegio,dc=int
adding new entry: ou=Idmap,dc=colegio,dc=int
adding new entry: uid=Administrator,ou=People,dc=colegio,dc=int
adding new entry: uid=nobody,ou=People,dc=colegio,dc=int
adding new entry: cn=Domain Admins,ou=Group,dc=colegio,dc=int
adding new entry: cn=Domain Users,ou=Group,dc=colegio,dc=int
adding new entry: cn=Domain Guests,ou=Group,dc=colegio,dc=int
adding new entry: cn=Domain Computers,ou=Group,dc=colegio,dc=int
adding new entry: cn=Administrators,ou=Group,dc=colegio,dc=int
adding new entry: cn=Account Operators,ou=Group,dc=colegio,dc=int
adding new entry: cn=Print Operators,ou=Group,dc=colegio,dc=int
adding new entry: cn=Backup Operators,ou=Group,dc=colegio,dc=int
adding new entry: cn=Replicators,ou=Group,dc=colegio,dc=int
adding new entry: SambaDomainName=COLEGIO,dc=colegio,dc=int
Please provide a password for the domain Administrator:
Changing password for Administrator
New password :
Retype new password :
```

prestaremos atención a donde buscamos los usuarios (passwd), las contraseñas (shadow) y los grupos (group) verificando que además de en los ficheros (files) lo hace en el directorio (ldap).

```
passwd:     files ldap
shadow:     files ldap
group:        files ldap
```

El fichero /etc/ldap.conf es el utilizado por pam_ldap y las bibliotecas NSS para la autenticación y la resolución de nombres, revise el Listado 2 para su configuración.

Resaltar del Listado 2 la línea en negrita, donde repetimos la entrada para nss_base_passwd apuntando a la rama de los equipos del dominio, ya que Samba3 necesita de nss_ldap para encontrar el uid+gid de las máquinas del dominio, podíamos haber resuelto esto también colocando los equipos dentro de la rama de usuarios.

Para comprobar que todo está funcionando podemos ejecutar *getent passwd* y *getent group* para ver que consultamos el directorio a la hora de obtener los usuarios y los grupos.

Configuración de Samba

Nos toca decirle al servidor Samba que busque a los usuarios en el directorio y que se comporte como un PDC de Windows. El fichero de configuración de Samba se encuentra en */etc/samba/smb.conf*. Debemos modificar la configuración de Samba para que se asemeje a la que aparece en el Listado 3.

Una vez configurado el servicio, ejecute testparm (herramienta para testear la configuración perteneciente a la suite de Samba) prestando atención a las siguientes entradas Server role: ROLE_DOMAIN_PDC y por supuesto a Loaded services file OK que verifica que el archivo es correcto.

Creamos los directorios necesarios como aparece en el Listado 4.

El único parámetro que no aparece en el fichero de configuración y que es necesario para que el administrador pueda llevar a cabo las operaciones necesarias sobre el servidor, es la contraseña que le permita autentificarse.

Creamos el password del usuario Manager en la bbdd de Samba:

```
[root@lobezno Samba]# smbpasswd
 -w 12345
Setting stored password for
"cn=Directory Manager" in
secrets.tdb
```

Interactuando con el directorio: smbldap-tools

En este punto tenemos 389 directory server y Samba perfectamente configurados para asumir el rol de PDC.

En primer lugar obtenemos el SID de Samba para el PDC:

```
[root@lobezno etc]# net getlocalsid
SID for domain LOBEZNO is: S-1-5-21-
XXXXXXXXX-XXXXXXXXX-XXXXXXXXX
```

Si obtenemos una respuesta como la de arriba es que vamos por el buen camino, hasta ahora no hemos iniciado todavía el servicio de Samba y no debe estar iniciado para conseguir el SID.

Ahora edite el fichero */etc/smbldap-tools/smbldap_bind.conf* para colocar las credenciales para el acceso de smbldap-tools al directorio:

```
slaveDN="cn=Directory Manager"
slavePw="12345"
masterDN="cn=Directory Manager"
masterPw="12345"
```

Continúe modificando el fichero de configuración principal de smbldaptools: */etc/smbldap-tools/smbldap.conf* (vea Listado 5) para configurar cómo interactúa smbldap-tools con nuestro directorio.

Listado 7. Salida net groupmap list

```
[root@lobezno ~]# net groupmap list
Domain Admins (S-1-5-21-1375073545-2757690610-XXXXXXXXX-512)
   -> Domain Admins
Domain Users (S-1-5-21-1375073545-2757690610-XXXXXXXXX-513)
   -> Domain Users
Domain Guests (S-1-5-21-1375073545-2757690610-XXXXXXXXX-514)
   -> Domain Guests
Domain Computers (S-1-5-21-1375073545-2757690610-XXXXXXXXX-515)
   -> Domain Computers
Administrators (S-1-5-32-544) -> Administrators
Account Operators (S-1-5-32-548) -> Account Operators
Print Operators (S-1-5-32-550) -> Print Operators
Backup Operators (S-1-5-32-551) -> Backup Operators
Replicators (S-1-5-32-552) -> Replicators
[root@lobezno ~]#
```

Listado 8. Zona DNS

```
root@lobezno ~]# cat /var/named/colegio.int.hosts
$ttl 38400
colegio.int.     IN       SOA      lobezno. administrator.colegio.int.
(
                          1256845598
                          10800
                          3600
                          604800
                          38400 )
colegio.int.    IN      NS      lobezno.
lobezno   A 192.168.1.200
malaspina A 192.168.1.201
_ldap._tcp.dc._msdcs  SRV 0 100 389 lobezno
_ldap._tcp.pdc._msdcs  SRV 0 100 389 lobezno
IN 1H SRV 0 100 389 lobezno.colegio.int
_ldap._tcp  SRV 0 100 389 lobezno
_ldap._tcp.colegio.int-site._sites  SRV 0 100 389 lobezno
_ldap._tcp.colegio.int_site._sites.dc_msdcs  SRV 0 100 389 lobezno
_ldap._tcp.gc._msdcs  SRV 0 100 389 lobezno
_gc._tcp  SRV 0 100 3268 lobezno
_gc._tcp.colegio.int-site._sites  SRV 0 100 3268 lobezno
```

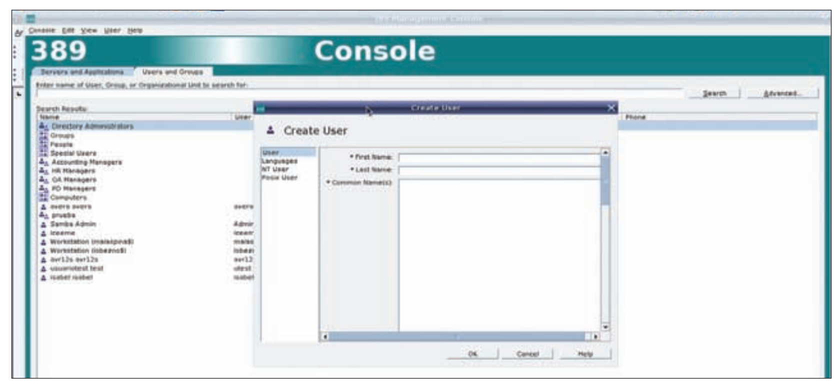


Figura 3. Alta de usuarios

Es el momento de "poblar" nuestro directorio, arranque los servicios de Samba y de directorio, y ejecute *smbldap-populate -a Administrator* (detallamos la salida del comando en el Listado 6). Revise que todo ha sucedido como esperábamos dando un vistazo al mapeo de grupos (Listado7).

Ahora es el momento de crear una cuenta para un usuario del dominio: *smbldap-useradd -m -a antonio* con el modificador -a aclaramos que es un usuario de Windows y con -m le creamos un directorio home, para más información consulte la página man del comando. Establezca también el password del usuario *smbldap-passwd antonio.*

Configuración de DNS (bind9)

Al igual que cuando configuramos un dominio con Active Directory necesitamos un servidor DNS en nuestra red para permitir a los clientes Windows unirse al dominio. BIND es el servidor de nombres de dominio más popular en Internet, y el más usado en plataformas Linux, se caracteriza por su flexibilidad y seguridad. La instalación mediante yum se reduce a ejecutar en una consola como root *yum install bind*. El servidor de nuestra red resolverá los nombres de nuestro dominio interno colegio.int y reenviará las peticiones de otros dominios a los servidores DNS de nuestro proveedor. La configuración de este servicio se sale del objetivo de este artículo, en su auxilio Fedora le ofrece una herramienta gráfica para su configuración *system-config-bind* que puede instalar vía yum: *yum install system-config-bind.* Ahora detallamos (vea el Listado 8) las modificaciones que debe realizar en la configuración de su zona para que los clientes Windows "conozcan" a nuestro servidor como un controlador de dominio a la hora de iniciar sesión.

Integración con Postfi

La documentación oficial de Postfix [6] es un buen punto de partida para la integración de nuestro directorio con un servidor de correo. Solamente añadir que 389-ds también es algo particular a la hora de integrarse en el correo y por compatibilidad con el antiguo directorio de Netscape usa el Objeto *mailgroup* en lugar del usual *CourierMailAccount*, para ampliar puede dirigirse al correspondiente apartado en el wiki de Fedora[7].

Configuración del cliente Windows

El procedimiento para unir la máquina Windows al dominio es el mismo que realzaría si el controlador de dominio fuera un Windows 2003. Suponemos que el cliente Windows tiene esta configuración de red:

IP : 192.168.1.201
Gateway: 192.168.1.1
Mascara de Red: 255.255.255.0
DNS: 192.168.1.200
Dominio de Búsqueda: colegio.int

Los pasos a seguir serían:

- Ingrese en la estación de trabajo como administrador,
- Botón derecho en "Mi Pc", clic en "Propiedades",
- Clic en la pestaña "Nombre de Equipo",
- Clic en el botón "cambiar",
- Seleccione "Dominio" y escriba "COLEGIO",
- Clic en el botón Aceptar,
- En la caja que aparece introduzca el usuario "Administrator" y la clave.

Conclusiones

En el wiki del proyecto 389 ya lo avisan[5], la integración entre Samba y 389-ds "al estilo fedora" es muy básica y todavía es un borrador, si quiere una integración completa y uso de Samba para manejar un dominio es mejor usar las herramientas de migración de idealix (smbldap-tools).

Fedora, en agosto de 2007, consciente de las deficiencias en la integración[5] comenzó con el desarrollo de las fdstools[8] para manejar el directorio y la correcta integración con Samba, POSIX, AIX, resolviendo las deficiencias de integración entre Samba y 389 Directory Server. El desarrollo se encuentra todavía en versión alfa, sólo usuarios con mucho tiempo libre o los desarrolladores pueden hacer uso de ellas.

Después de implementar en nuestro laboratorio las dos soluciones propuestas no tenemos más que dar la razón a los chicos de Fedora y usar smbldap-tools para integrar Samba con 389-ds.

Probar 389-ds nos ha permitido acercarnos a las soluciones corporativas de Red Hat: IPA server y Red Hat Directory Server, descubriendo que tienen algunas ventajas destacables frente al uso de OpenLdap estándar:

- La replicación "multimaestro" y la resolución de conflictos automática, es una cualidad que hasta la aparición de la rama 2.4 de OpenLdap no teníamos la oportunidad de usar.
- La consola de administración y la posibilidad de hacer las tareas de mantenimiento del directorio "sin parar" nos parecen mejoras muy acertadas.
- La posibilidad de sincronización con un AD de nuestra propia red.

Sin embargo estas ventajas quedan un poco eclipsadas, por la particular forma de implementar el directorio, tal vez su "extraña licencia", que provoca la falta de documentación fuera del entorno Red Hat.



En la red

- [1] *http://directory.fedoraproject.org/wiki/Install_Guide*
- [2] *http://es.wikipedia.org/wiki/Yellow_dog_Updater,_Modified*
- [3] *http://fferrer.dsic.upv.es/cursos/Integracion/html/ch06.html*
- [4] *http://directory.fedoraproject.org/download/ol-schema-migrate.pl*
- [5] *http://directory.fedoraproject.org/wiki/Howto:Samba#Samba_.26_Fedora_Directory_Server_Integration*
- [6] *http://www.postfix.org/LDAP_README.html*
- [7] *http://directory.fedoraproject.org/wiki/Howto:Postfix*
- [8] *http://fdstools.sourceforge.net/*

Car Arena

Comenzamos la sección de juegos de esta semana con un juego de coches y que como podéis ver en el título, compiten en una “Arena”, es decir un circuito cerrado al más puro estilo “Micro Machines”. Podríamos decir que es un juego inspirado en el clásico “Micro Machines” pero llevado a las tres dimensiones y renovado en todos los aspectos. Lo que más me impresionó del juego en un primer momento fueron sus gráficos, que además de tridimensionales tienen una apariencia poco seria, en el buen sentido de la palabra, lo que va muy en línea con el juego en sí. El problema es que la cámara que sigue a los coches, no siempre enfoca el ángulo mejor para cada momento, por lo que en ocasiones perderemos un poco de vista la acción y tendremos problemas si estamos en pleno adelantamiento. El juego pone a nuestra disposición cinco vehículos distintos y cinco circuitos. Los vehículos



Figura 1. Car Arena

son: Mitsubishi Lancer Evo 7, Subaru Impreza, Lancia Delta Evolution, Peugeot 206 y Fiat Punto S200. En realidad en el juego no se pone la marca porque no tienen los derechos, pero sí se indica el nombre y por la forma y colores podéis identificarlos. Cada uno de ellos tienen sus propias características y comportamiento.

Por supuesto, el juego también permite partidas multijugador a traés de la red o individuales. Si elegimos jugar nosotros contra el ordenador, tendremos dos niveles de dificultad. También existe un modo de juego llamado “Free Ride” en el que llevamos nuestro vehículo por el circuito en solitario para saber sacarle el máximo partido en cada situación. Los circuitos aunque son sólo cinco, también son muy variados y merece la pena probar todos, porque tienen detalles muy curiosos como puentes a distintas alturas. Gráficamente deciros que se ha diseñado con Blender, el fantástico software libre de dibujo tridimensional. Lo que sí tengo que deciros es que el juego no es libre. Sí es gratuito pero su código es cerrado. Aún así, creo que será interesante para todos para poder echar un buen rato. Os advierto que aunque no tengamos a nadie con quein jugar, el modo en solitario contra la máquina es muy divertido. Existen instalables también para Windows. Para Linux hay un fichero comprimido con los binarios. Como podemos comprobar, no todos los juegos que hay para Linux son libres, y existen además éstos, que son gratuitos, que siempre vienen mejor que los de pago.

En conclusión, un juego para echar un rato muy divertido en un momento. Buenos gráficos, mucha velocidad y una jugabilidad adecuada, nos proporcionará muchos ratos de evasión de lo más divertido.

http://www.izedesign.it/cararena/index_eng.html



World Of Goo

El segundo juego de este mes tampoco es libre, ni tampoco gratuito, pero merece la pena incluirlo en esta sección. Su nombre es World Of Goo y es un juego un tanto curioso por muchas circunstancias. Lo primero deciros que se trata de un juego del género de los puzzles, pero que está muy centrado en el campo de los efectos físicos sobre el juego. Un juego no demasiado corriente. Antes de seguir comentaros respecto al precio, que aunque es de pago, a lo largo de su vida he visto por lo menos dos „ofertas” de gran interés para su compra. La primera ofrecía el juego por 1 céntimo y la segunda te pedían los autores que les pusieras un valor al mismo. Como veis, es curioso hasta para la venta. Sigamos.

El juego no es sólo para Linux, ni siquiera para PC, sino es un juego de Wii (esto tiene muy buena pinta) que también tiene versiones

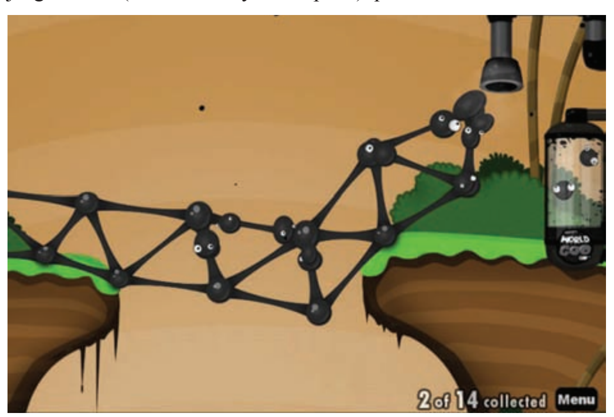


Figura 2. World of Goo

para los tres grandes sistemas operativos del ordenador de escritorio: GNU/Linux, Mac OS X y Microsoft Windows. Está producido y desarrollado por 2D Boy, una empresa formada por dos antiguos empleados de Electronics Arts, y en su realización, además de él, colaboró otra persona más, lo que hace un total de tres personas desarollando el juego (una cifra muy baja para ser una producción comercial).

Las herramientas utilizadas para su desarrollo son prácticamente todas libres. Podemos destacar la librería/API SDL (Simple DirectMedia Layer), Open Dynamics Engine y TinyXML, para desarrollo y Subversion y Mantis Bug Tracker para coordinación en el proceso de creación.

El objetivo del juego es ir componiendo caminos basados en bolas “Goo”, cada una con sus características especiales, para llegar hasta el final. La física juega un papel muy importante en todo el desarrollo del juego: las fuerzas, incluida la gravedad y las condiciones climáticas. Las bolas “Goo” son de diferentes tipos y aunque algo muy extraño, hacen gestos y emiten sonidos. La ambientación del juego al completo, incluido los gráficos, recuerdan a las películas de Tim Burton. La música, es otro detalle muy cuidado y va a la perfección con el juego. Para jugar, podéis hacerlo con prácticamente cualquier equipo no demasiado antiguo. No requiere una aceleradora independiente ni ninguna tarjeta gráfica muy potente.

La verdad es que es un juego del que es difícil transmitir mucho más a través de unas líneas. Es de lo más curioso que he visto últimamente y por lo que veo, no soy el único al que resulta divertido. Están desarrollando versiones incluso para dispositivos como el iPhone.

http://www.worldofgoo.com/



Linux en el instituto

Antonio Gómez

En un centro educativo, resulta muy atractiva la idea de un ordenador central que cumpla funciones web, de filtrado de contenidos no aptos para menores y que permita centralizar y coordinar muchas actividades de enseñanza-aprendizaje basadas en las TIC.



linux@software.com.pl

Vamos a tratar de reflejar, a lo largo de dos artículos, la experiencia del Instituto de Educación Secundaria Eduardo Valencia, en Calzada de Calatrava (Ciudad Real) en lo referente a la reinstalación de la red informática del centro, centralizándola en la figura de un ordenador servidor con Ubuntu Server 9.04. Las funciones principales de dicho servidor dentro de la red serán:

- Racionalización de la conexión compartida a Internet de los equipos del centro, así como la posible denegación de acceso de ciertos equipos a determinadas direcciones, si así lo solicita el profesorado del centro(Squid).
- El trabajo como servidor web con Apache2.
- Facilitar la compartición de carpetas entre equipos Windows, Molinux y Ubuntu con Samba.
- Disponer de un sistema de correo interno (externalizable) que posibilite el intercambio de documentación entre el personal del centro.

Las razones para llevar a cabo un experimento tan arriesgado (en la medida en que durante el tiempo que duran los distintos experimentos de instalación, testeo, reinstalación, etc... mantenemos interrumpida parte de la normal actividad informática dentro del centro), son muchas y muy variadas; mencionemos sólo las más importantes:

- Incluso un instituto pequeño como el nuestro puede llegar a mantener más de 90 equipos conectados a Internet, compartiendo una conexión de apenas 3 Mb (muchos días, sólo en teoría). La presencia de un servidor proxy en el sistema que guarde en su caché las páginas más solicitadas aceleraría el proceso. Si tenemos en cuenta que además, en un centro educativo, se solicitan muchas veces las mismas direcciones web (varias páginas de corte educativo, o el portal de la Junta de Comunidades de Castilla la Mancha, por poner sólo dos ejemplos), el aumento en la velocidad de conexión resultará mucho más visible.
- Es muy interesante, a todos los niveles, disponer de un servidor web accesible, al menos en parte, a profesores

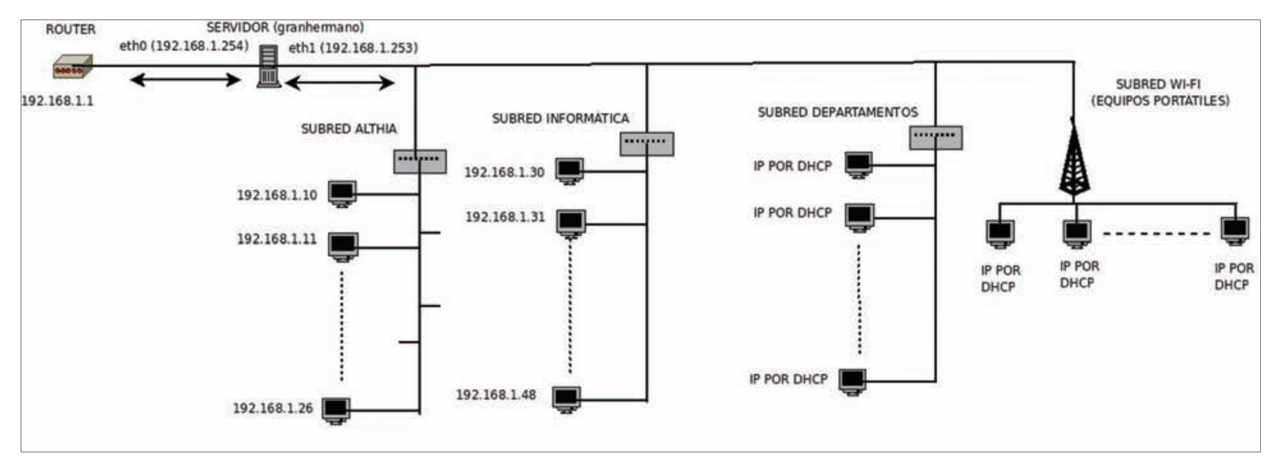


Figura 1. Configuración original de la red

y alumnos, con la autonomía propia que concede la presencia en el instituto de un servidor dedicado. Apache2 está demostrando ser muy potente y sencillo de configurar.

- Hay muchos equipos con Linux (Ubuntu y Molinux) instalados en el centro (de hecho, el último año, la Junta de Comunidades de Castilla la Mancha dotó a todos los profesores de la Comunidad Autónoma con un ordenador portátil con arranque dual en Windows XP y Molinux Hidalgo), pero el sistema de Microsoft sigue siendo aún mayoritario. Es muy necesario facilitar la cohabitación de ambos sistemas en el intercambio de archivos entre los distintos miembros de nuestra comunidad. Samba resolverá este conflicto.
- El correo electrónico es una herramienta muy presente hoy en día en las distintas actividades burocráticas y de gestión administrativa que conlleva el día a día en un centro educativo. Hemos dotado a cada Departamento y al Equipo Directivo del centro con sus propias direcciones web, que evitan al docente tener que utilizar su dirección privada de correo (que no tiene por qué verse obligado a utilizar en estos avatares), y dotan a nuestra institución de una pátina de autonomía y respetabilidad que supondrán una mejor imagen externa.

En efecto, para el redactor de este artículo son razones más que suficientes para iniciar una pequeña "guerra interna" a lo largo de dos o tres semanas con sus compañeros de profesión; una sucesión de pequeños conflictos que, me alegra decirlo, no pasaron del rango de pequeñas reclamaciones, casi siempre por la no disponibilidad de la red en varias ocasiones, lo que provocó unas cuantas veces la interrupción de determinadas actividades de aprendizaje que algún profesor llevaba a cabo con su grupo (realización de webquests, trabajo con blogs,...). Desde aquí mi agradecimiento a todos mis compañeros por la paciencia que demostraron en todo momento. Este tipo de situaciones refuerzan siempre nuestra sensación de auténtica pertenencia a una comunidad.

Instalación física del servidor

El sistema elegido finalmente para nuestro ordenador central ha sido Ubuntu Server 9.04, dada su sencillez, fiabilidad y eficiencia. Además, ya dispone de la mayoría de los paquetes necesarios para iniciar nuestra actividad, y sus repositorios oficiales ya disponen de la mayoría de los que aún no estarán instalados pero necesitaremos. Recordemos que la herramienta más adecuada para instalar un paquete desde consola será aptitude. Ej:

```
# aptitude install openssh.
```

Bien. No es objeto de este texto indagar sobre la instalación de un sistema operativo Linux en un equipo, además de que en la actualidad se ha convertido en una actividad de lo más sencilla. Recordemos que Ubuntu Server, como todos sus homónimos, no dispone en un principio de un entorno de escritorio, ni Gnome ni Kde, dado que no está pensado para equipos personales, y su presencia sólo redundaría en una menor eficiencia. Así que será imprescindible tener un cierto nivel de manejo con la shell.

Nuestro equipo en cuestión dispone de dos tarjetas de red, eth0 (conexión al router) y eth1 (a la red local). Su nombre de equipo, en un alarde de humor negro, será *granhermano*, y el nombre de usuario con posibilidades de root *jefazo*.

La red original en la que queremos implantar este equipo estaba configurada del siguiente modo: la conexión al exterior se hacía sobre un router, que alimentaba a cuatro subredes locales:

Aula Althia: sala con dieciséis ordenadores con arranque dual Windows y Molinux, parte de un proyecto de la JCCM de hace un par de años, para mejorar la informatización de los colegios e institutos. Aula de informática: sala con dieciocho ordenadores con arranque dual Windows y Ubuntu.



Figura 2. Explorando un poco, se puede encontrar rápidamente la función que necesitamos

Departamentos Didácticos: desde un switch, se cableó a lo largo de todo el centro el acceso a Internet del ordenador de cada Departamento. Unos veinte ordenadores más, contando los tres de la biblioteca del instituto.

Red Wi-Fi. Desde hace dos años, la Junta de Comunidades dotó también de los recursos necesarios para garantizar el acceso wi-fi a cualquier ordenador desde cualquier punto del instituto. A la sazón, tenemos instalada la red correspondiente de puntos de acceso por todo el edificio.

La inserción de nuestro servidor en este conglomerado nos dejaría en una situación como la de la Figura 1.

Los pasos que daremos, una vez instalado físicamente el servidor dentro de la red, serán los siguientes:

- Actualizar adecuadamente el servidor, y añadir los usuarios (departamentos didácticos y profesores) que deseamos tengan acceso al servidor.
- Configurar el sistema de acceso remoto al equipo *granhermano* mediante consola (*OPEN SSH*) y explorador web (*WEBMIN*).
- Preparar el sistema de carpetas compartidas mediante *SAMBA*.
- Configurar el servidor proxy *SQUID*, que deseamos que funcione en modo transparente. Para ello, también tendremos que configurar un servidor *DHCP* e *IPTABLES*.

Hasta aquí, lo que trataremos a lo largo del presente artículo. En el próximo número, trataremos también la instalación y configuración de *APACHE2*, así como la creación de un servicio interno (accesible desde el exterior) basado en *POSTFIX*, *DOVECOT* y *SQUIRRELMAIL*, así como la posibilidad de dar acceso limitado al servidor a los profesores para realizar tareas personales sencillas, como cambiar su propia clave de acceso.

Bien, vayamos a ello. La historia no la escriben los cobardes (aunque bien es cierto que éstos, al menos, tienen la oportunidad de leerla después).

Configuración del acceso remoto del servidor

De acuerdo. Hemos instalado Ubuntu Server en nuestro equipo. De manera provisional, mientras duren los testeos, el equipo dispone de los periféricos mínimos (teclado y monitor), pero en la instalación definitiva, no será así, dado que estará debidamente retirado de la vista al público. Por ello, el primer paso es asegurar que podemos acceder a *granhermano* tanto dentro como fuera de nuestra red LAN. Para ello, instalaremos *OPENSSH* para el acceso remoto por terminal, y la utilidad *WEBMIN* para el acceso gráfico vía web.

Listado 1. Archivo /etc/network/interfaces para configurar dos tarjetas de red, eth0 y eth1

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
        address 192.168.1.254
        gateway 192.168.1.1
        netmask 255.255.0.0
        network 192.168.1.0
        broadcast 192.168.1.255
     auto eth1
iface eth1 inet static
address 192.168.1.253
netmask 255.255.0.0
```

Listado 2. Configurando el servidor DHCP /etc/dhcp3/dhcpd.conf

```
#Opciones generales
#Establecemos como servidor dns primario el propio router, que
normalmente estará configurado #para funcionar como tal.
option domain-name-servers 192.168.1.1;
#Broadcast-address fija la máscara que se utilizará para distribuir la
señal (toda la red)
option broadcast-address 192.168.255.255;
#Fijamos como puerta de enlace primaria la dirección IP de eth1
option routers 192.168.1.253;
#Fijamos como nombre de dominio el de nuestro servidor
option domain-name "granhermano";
#Desactivamos la actualización automática de dns dinámicas.
ddns-update-style none;
#La opción authoritative marca nuestro servidor como el principal
proveedor DHCP de direcciones; #si comentamos esta opción, deberíamos
desactivar el servidor DHCP de nuestro router;
authoritative;
# Creamos nuestra subred
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.50 192.168.1.200;
        default-lease-time 21600;
        max-lease-time 43200;
}
```

Configuración del router para el acceso remoto

Dependiendo del modelo de router que se utilice en cada red, los pasos de configuración a seguir pueden variar, pero siempre se vertebrarán en torno a un protocolo parecido a éste:

- El router puede configurarse mediante nuestro explorador web favorito (*Firefox*, por ejemplo), normalmente en la dirección 192.168.1.1 (naturalmente, necesitaremos el nombre y la clave de usuario autorizado).

Tabla 1. Redireccionamiento de puertos web en el router

PUERTO	DIRECCIÓN IP
8080	192.168.1.1
80	192.168.1.254 (eth0 en *granhermano*)

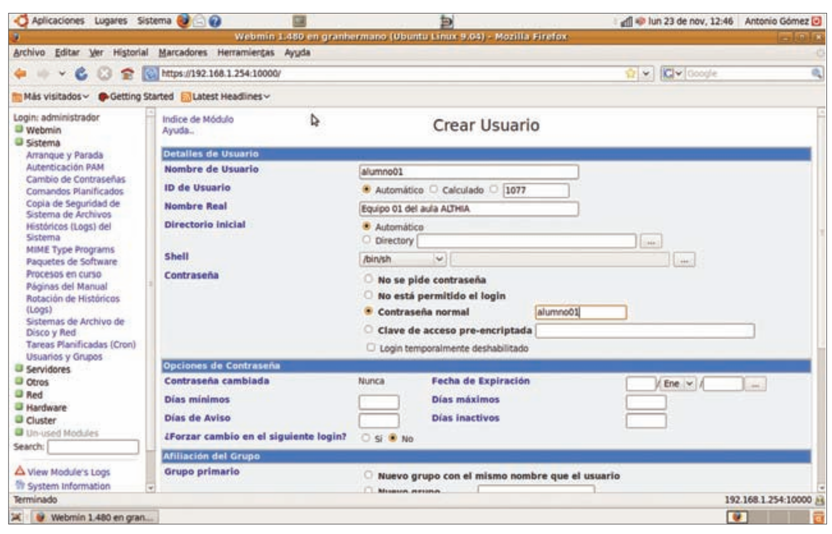


Figura 3. Creando usuarios y grupos desde WEBMIN

- Necesitamos indicar al router a qué ordenador debe desviar las peticiones externas realizadas a determinados puertos. En nuestro caso, debemos reconducir a la IP 192.168.1.254 (eth0) todas las peticiones hechas al puerto 22 (ssh), 10000 (webmin), 80 (apache) y 25 (postfix). Una opción para evitar problemas los primeros días sería aprovechar la característica DMZ (Demilitarized Zone) que suelen ofrecer los router estándar, y que abren directamente todos los puertos de un ordenador en particular dentro de la red, si bien es recomendable limitarse a los puertos señalados una vez pasado el período de pruebas.
- En nuestro caso particular, y para evitar las molestias derivadas de los cambios periódicos que pueden derivarse de la utiliza ción de IP's dinámicas, recurrimos al servicio gratuito NO-IP, que nos permite recurrir a sus servicios de DNS y contar con nuestra propia URL (*http:/eduardovalencia.no-ip.org)*.

Configuración del servicio de DNS dinámicas

Como ya sabemos, lo que conocemos como direcciones web o URL son transcripciones a lenguaje "normal" de las direcciones IP que corresponden a las ubicaciones físicas de los servidores web cuyos servicios deseamos utilizar. Los responsables de estas relaciones entre IP y URL son los servidores DNS. Tener nuestro servidor localizable en Internet nos ofrecía dos problemas.

- Necesitamos elegir dicho servidor DNS, en nuestro caso de carácter gratuito. Optamos por NO-IP (*http://www.no-ip.com*).
- Nuestro servidor se encuentra dentro de una red local LAN. El servidor DNS mantendrá en su base de datos la dirección IP general (la de nuestro router), que deberá redireccionar las peticiones web a *granhermano*, pero teniendo en cuenta, además, que la política de los actuales proveedores de ADSL considera una práctica aconsejable el cambio de dirección IP en los router de manera periódica. Deberemos, pues, asegurarnos de que cada vez que dichos cambios de dirección se produzcan se comunique a la base de datos de nuestro DNS, lo que se logra mediante la instalación de un programa cliente.

La ventaja de estos problemas estriba en que conocemos, por lo menos, su naturaleza, así que procedemos a resolverlos.

a) Daremos por supuesto que estamos registrados en uno de estos servicios (*dyndns* y *no-ip.com* son los más populares), de modo que tenemos una dirección asignada, y hemos descargado ya el programa cliente que avisa a *dyndns* o *no-ip* del cambio en la IP general cuando éste se produce. Dicho programa se puede descargar de la página web correspondiente, y habría que descomprimir la carpeta, compilar y configurar. El archivo suele incluir una sección de documentación, donde algún fichero del tipo LEEME.PRIMERO nos indicará cómo debemos actuar. Los S.O. de base Debian (nuestro queridísimo Ubuntu entre ellos), incorporan ya, en sus repositorios, dicho paquete: `# apt-get install no-ip` o `# aptitude install no-ip`.

Sólo restaría configurarlo (se nos pide la dirección de correo y la contraseña con la que constamos como usuarios registrados en dicho servicio).

b) Normalmente, accedemos a la utilidad web de configuración del router en la dirección 192.168.1.1, previa identificación como administrador de la red LAN (esta identificación no tiene nada que ver con el usuario *jefazo* de nuestro equipo). En un principio, la mayoría de los router incorporan directamente una opción de configuración de IP dinámica (por ejemplo, en los de marca COMTREND aparecen en el menú: *DNS->Dynamic DNS,* o en los de tipo *U.S. ROBOTICS* se ofrece en *DNS->DNS Dinámicas)*. En esa opción debería pedir nuestra identificación como usuarios del servicio (es decir, habría que repetir el paso anterior, pero en el router en vez de en el servidor).

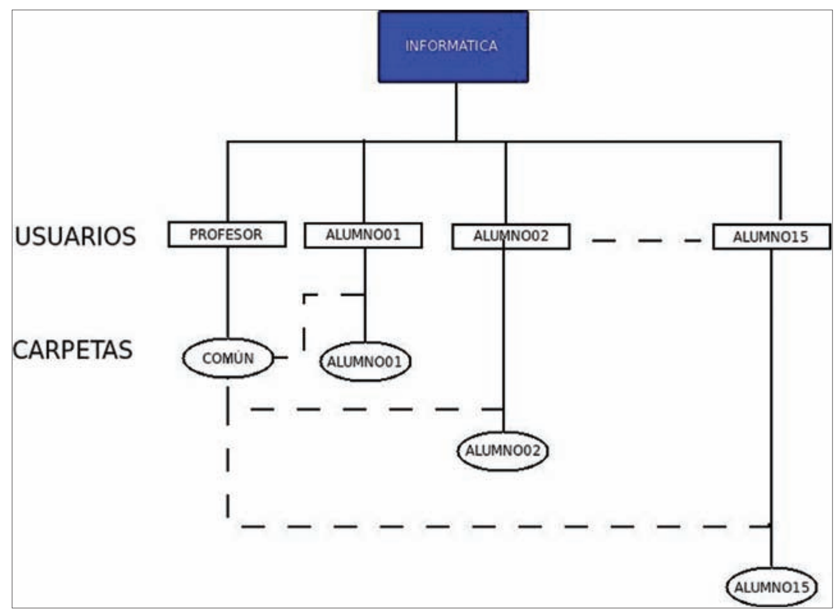


Figura 4. Estructura del grupo de trabajo INFORMÁTICA

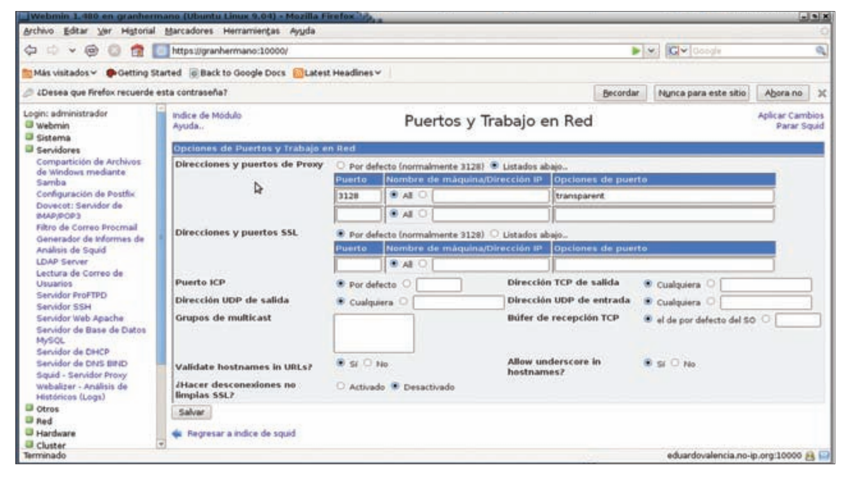


Figura 5. En WEBMIN, podemos configurar nuestro servidor Squid como transparente

c) El tercer paso es donde tropezamos varias veces, antes de caer en lo obvio: nuestro router está preparado para recibir peticiones del puerto 80 desde los servidores dns de *dyndns* o *no-ip*, *granhermano* está preparado para avisar a dichos servidores de los cambios que se puedan producir en la IP del router (aunque creo, en el fondo, que este segundo aspecto suele ser totalmente inútil en el conjunto); pero... cuando se hace una petición a nuestro router, ¿Cómo sabe el router a qué ordenador dirigir la petición web dentro de nuestra red? Por eso, tenemos que configurar el router para que envíe cualquier petición que se le haga, al puerto 80 por ejemplo, en el caso del servidor web, a la dirección IP de nuestro *APACHE*. Investigando en Internet, descubrimos que un problema añadido en varios casos es que el router puede tener reservado dicho puerto 80 a su propia página web de configuración. La solución que se propone consistiría, si eso nos da problemas, en cambiar el acceso a dicha web interna al puerto 8080, y dejar el 80 a tu APACHE. En nuestro caso, quedaría algo como en la Tabla 1.

Instalación de *OPENSSH*

Esta utilidad nos permitirá gobernar de manera remota el ordenador desde cualquier otro equipo que disponga de un programa cliente que permita la conexión, sea Windows (*PUTTY es la mejor opción para mí), o Linux (desde consola, SSH-CLIENT),* con la ventaja añadida de que el intercambio de información entre equipos se realiza de manera encriptada, lo que aumenta la seguridad.

Los pasos a seguir serán los siguientes:

- Instalación del servidor: *aptitude install openssh-server.*
- Instalación de la utilidad cliente, tanto en el servidor como en nuestro equipo personal:*aptitude install openssh-client.*
- En el servidor, configuración de la utilidad: *ssh localhost.*

Este tercer paso implica la creación de un par de claves (keys), una pública y una privada, cuya combinación entre nuestro equipo personal y el servidor asegurará nuestra identificación adecuada en cada conexión como usuario autorizado.

Para comprobar si todo ha funcionado, bastará con abrir una consola en nuestro equipo personal (en el caso de Microsoft, iniciar *PUTTY*), y tratar de conectarnos: `# ssh jefazo@granhermano` (dentro de la red local) o `# ssh jefazo@IPPUBLICADENUESTRARED` (por ejemplo, desde nuestro domicilio, siempre que hayamos configurado correctamente el router).

Si todo ha ido bien, aparecerá un mensaje indicando la comprobación de la *public key* y pidiendo confirmar la conexión, hecho lo cual se nos pedirá la contraseña del usuario *jefazo*, que tiene los permisos necesarios para acceder al sistema.

Instalación de WEBMIN

Webmin es una utilidad escrita en PHP que nos permite la gestión remota de equipos con Linux instalado a través de una interfaz gráfica que facilita, no mucho, sino muchísimo, la gestión de un equipo cuando somos usuarios poco avezados, y por qué no decirlo, un poco sobrepasados por nuestra falta de experiencia.

Partimos de que en este momento, nos encontramos en nuestra carpeta *HOME*. Vamos a descargarnos el paquete mediante la herramienta *wget*, le daremos los permisos necesarios al archivo descargado, un autoinstalable *.deb, y finalizaremos instalándolo:

```
# wget http://prdownloads.sourceforge.net/webadmin/webmin_1.490_all.deb
# chmod 755 webmin_1.490_all.deb
# ./webmin_1.490_all.deb
```

Para comprobar la funcionalidad de esta interfaz, nos conectaremos desde el explorador web de nuestro equipo personal: *https://192.168.1.254:10000* desde dentro de la red local, aunque también podría servir: *https://granhermano:10000.*

La dirección de acceso externo sería: *https://IPPUBLICADENUESTROROUTER:10000.*

El correcto desarrollo de esta fase de nuestro trabajo permitirá el acceso desde cualquier equipo, dentro o fuera de la red, al servidor central. Los prudentes y los cobardes (métaseme en el grupo que más agrade a nuestro dignísimo lector), omitirán la desconexión de teclado y monitor del equipo mientras no se demuestre la estabilidad del conjunto por activa y por pasiva, pero está claro que es algo que ya podría hacerse.

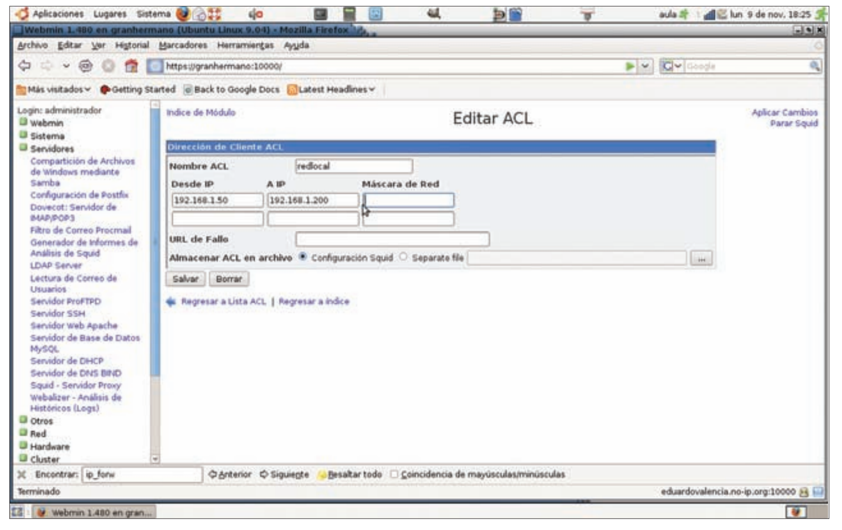


Figura 6. En el ejemplo, definimos la red con acceso a Internet desde 192.168.1.50 a 192.168.1.200

Listado 3. Contenido del archivo configurandoiptables.sh

```
#Habilitamos el ruteo
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
#Redireccionamos todos los paquetes que vienen por el puerto 80 a eth1
hacia el paquete 3128
sudo iptables -t nat -A PREROUTING -i eth1 -s 192.168.1.0/24 -d !
192.168.1.0/24 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

En ocasiones, después de instalar nuevas utilidades, es posible que echemos en falta el correspondiente módulo en el menú de *WEBMIN*. Es posible que esta interfaz web no sepa cómo manejar esta aplicación, pero eso sólo sucederá en contadas y muy específicas situaciones. Lo más normal es que baste con *Refresh Modules* (si es que todavía no hemos cambiado el lenguaje por defecto, *WEBMIN ->Webmin->Change Language and Theme*; por cierto, ¿sabía el lector que puede, incluso, operar en lenguaje Klingon?).

Configuración de las tarjetas de red

Durante la instalación de nuestro sistema operativo, en un momento determinado se nos preguntará sobre la configuración de nuestra red. En ese momento, por razones de comodidad, dejaremos que el propio programa configure dichas tarjetas automáticamente, por DHCP, pero cuando terminemos, deberemos configurar dichas tarjetas en el archivo /etc/network/interfaces (el editor de textos *nano* o *vi* servirá de sobra). El contenido de dicho archivo debería ser (ver Listado 1).

Creación de usuarios y grupos en el sistema

Muy importante, de cara a otros pasos que se detallan después, es tener muy claro qué usuarios deberían tener acceso a *granhermano* en cada caso, así como en qué grupos, primarios y secundarios, querremos encuadrarlos, y qué permisos de lectura y escritura les vamos a asignar en cada servicio. Este es uno de los momentos en que nos sentiremos agradecidos a *WEBMIN* por su simplicidad, aunque esto menoscabe, en cierto modo, la filosofía del usuario de la shell de sacrificar simplicidad por potencia.

En un principio, la política de cualquier sistema con base UNIX es orientarse al trabajo con múltiples usuarios. Cada vez que alguien se conecta al ordenador, se identifica con su nombre y su clave, y el equipo identifica automáticamente cuál es su carpeta de trabajo (*/home/nombredeusuario*), donde tiene permisos de lectura y escritura, así como el grupo o grupos a los que dichos usuarios pertenecen, y donde también pueden configurarse las atribuciones que deseemos asignarle. Recordemos que cada archivo y carpeta (como archivo "especial", que es como están considerados) incluyen un sistema de tres tipos de permisos (lectura, escritura y ejecución) para tres tipos distintos de usuarios (propietario, grupo de propietarios y otros), expresados por un sistema octal, en cuya base no entraremos ahora.

En cuanto a los grupos de usuarios de tipo "normal", lo cierto es que el administrador de la red no entró en excesivas consideraciones de tipo técnico ni organizacional: si los usuarios son alumnos y profesores, habrá un grupo *alumnos* y un grupo *profesores.*

Los usuarios individuales, a priori, se irían creando a petición de los distintos miembros de la comunidad que así lo deseen (siempre que el administrador lo considere conveniente, por supuesto). En un principio (como se verá en el apartado destinado a *SAMBA*), se crearán usuarios para cada ordenador individual de los espacios informáticos comunes (a la sazón, de *alumno01* hasta *alumno15* en el aula *ALTHIA*, y de *informatica01* hasta *informatica15* en el aula *INFORMÁTICA*, así como al ordenador de profesor de cada aula), y un usuario estándar por Departamento Didáctico (*tecnología, matemáticas, lengua, francés, inglés,...* etc).

Cada usuario tendrá permisos de lectura y escritura en su carpeta *home*, y podrá especificarse, en cada caso, si deseamos otorgar al resto de miembros del grupo permisos para escribir o ejecutar programas en dichas carpetas, o simplemente acceder a ellas. A la hora de intercambiar trabajos y documentos a través de la LAN, esta característica de los sistemas GNU/UNIX abrirá al profesor posibilidades muy interesantes. Así pues, entramos en nuestra interfaz *WEBMIN*, y una vez identificados como administradores con permisos de root, seleccionaremos la opción *Sistema->Usuarios y grupos->Grupos locales*. La pantalla de información, con los usuarios ya creados, que nos aparece, es de por sí suficientemente explicativa. Baste decir que crearemos los grupos de usuarios que consideremos necesarios (*Profesores* y *Alumnos*, como hemos dicho).

Del mismo modo, iremos creando cada uno de los usuarios individuales que consideremos necesario (*Sistema->Usuarios y grupos->Usuarios locales*), ateniéndonos siempre a las siguientes reflexiones:

- Los usuarios tienen un grupo primario (el que originalmente les está destinado), pero también se les puede asignar uno o más grupos secundarios.
- Todos los usuarios deberían tener una contraseña, por más simple que ésta sea.
- En casos especiales, pueden crearse usuarios sin carpeta *home*, o denegárseles el acceso a determinados servicios.

Compartición de carpetas y archivos con SAMBA

A continuación, pasamos a configurar el sistema SAMBA de cara a compartir archivos en red en S.O. Windows y Linux.

Organización de los grupos de trabajo

Muy bien, ya tenemos algo con lo que empezar. Si lo ya hecho hasta ahora no nos ha resultado lo suficientemente complicado, y a esta altura del artículo no nos hemos echado a llorar más de tres o cuatro veces, podemos enfrentarnos a *SAMBA*.

SAMBA (juego de palabras relacionado con SMB, *Server Message Block*, un protocolo de red diseñado para estas funciones) es una utilidad que permite compartir archivos entre varios ordenadores de una misma red local, independientemente del sistema operativo que éstos utilicen. Es altamente configurable, y trabaja no sólo con archivos, sino también con periféricos como las impresoras. Lo único que necesitamos tener claro a la hora de instalar samba es la organización de nuestros *GRUPOS DE TRABAJO.*

En el caso particular de nuestro instituto, nos interesa particularmente configurar dos grupos de trabajo: el del aula *ALTHIA* y el del aula *INFORMÁTICA*, dado que son conjuntos de ordenadores en los que interactuará el profesorado con varios grupos de alumnos, unas veces trabajando con Linux, otras veces no. Centrémonos en la estructura, por ejemplo, de *INFORMÁTICA* (Figura 4).

Disponemos de un usuario,*profesorinformatica*, que debería tener permisos de lectura y escritura en todas las carpetas. Por otro lado, cada uno de los otros quince ordenadores deberá atender a un usuario denominado *alumnoXX* (siendo XX el número del equipo), que deberá tener permisos de lectura y escritura en su propia carpeta, no debería tener acceso directo a las carpetas de red de sus compañeros (para evitar la eterna tentación del *copiar y pegar*),

así como permisos de sólo lectura en una carpeta, *COMUN*, donde el profesor pueda crear los documentos de base a partir de los cuales el alumnado empiece a trabajar, pero que no sean borrables desde los otros equipos, por evitar errores más que por otra cosa (no olvidemos que trabajamos con grupos de jóvenes que suelen ser muy heterogéneos; siempre hay algún chico o chica que cometerá la peor equivocación posible en el peor momento). Utilicemos este ejemplo como base de configuración de nuestro sistema de compartición.

Instalando y configurando *SAMBA*

Si no hemos instalado este paquete durante la instalación de *UBUNTU 9.04*, podemos hacerlo desde consola:

```
# aptitude install samba
```

Ante la pregunta que el sistema nos hace sobre si deseamos hacer funcionar SAMBA como daemon o desde inet-d, elegimos la primera opción .

La configuración de *SAMBA* puede hacerse de manera directa, desde el terminal, editando el archivo */etc/samba/smb.conf*. Sin embargo, como hemos remarcado varias veces a lo largo de este texto, nos estamos dirigiendo a lectores con un nivel de dominio de la shell bajo o incluso nulo, así que recurriremos a *WEBMIN*.

Una vez conectados a *WEBMIN*, y partiendo de que el usuario ya habrá configurado la herramienta a su gusto, idioma castellano incluido, desde el apartado S*ervidores,* subapartado *SAMBA*, deberemos dar los siguientes pasos:

- Creación de los usuarios en Ubuntu Server. Se puede hacer mediante terminal (comando *adduser)*, o desde *WEBMIN* (*Sistema->Usuarios y grupos)*.
- En *SAMBA*, *Crear una nueva compartición de archivo*. En el formulario subsiguiente, nos limitaremos a marcar la opción *Compartición de directorios de inicio*, y pincharemos en Aceptar, lo que nos generará la compartición *homes* (la carpeta home de cada usuario estará compartida por el protocolo SMB). Hay que recordar que en el home de *profeinformática* deberemos crear una carpeta denominada *COMUN.*
- *WEBMIN* incluye una opción para configurar automáticamente la sincronización de usuarios de UNIX y SAMBA, pero no parece funcionar muy bien. Para evitar errores, volveremos a añadir los usuarios de samba por la terminal, con el comando *smbpasswd nombreusuario -a*, tras el cual deberemos introducir la contraseña de dicho usuario.

Conectando los equipos WINDOWS a *SAMBA*

En cada ordenador del aula, en Windows, tendremos acceso a dichas carpetas creando una UNIDAD DE RED en MIPC (menú *Herramientas->Conectar a unidad de red).* El trayecto de cada carpeta será \192.168.1.253\alumnoXX y \192.168.1.253\profesor-informatica\comun.

El nombre de usuario y password que se solicitarán a continuación serán los del alumno en cuestión. ¡No debemos olvidar dejar marcada la casilla que recordará esos parámetros cuando el equipo se reinicie!

Servidor SQUID transparente, iptables, filtro web

Como siempre, empezaremos asegurando que disponemos de los paquetes necesarios. Si alguno nos faltara, no hay más que recurrir a nuestra "identidad secreta" como superhéroe *root*:

```
# sudo su aptitude install squid
squid-common
```

Squid es un servidor proxy caché, cuya principal función es regular los distintos accesos a Internet por parte de todos los equipos dentro de la red local. Su instalación y arranque son sencillísimos, especialmente desde *WEBMIN*.

Ahora bien, para que dicho proxy tenga efectividad, deberíamos reconfigurar nuestro navegador especificando la dirección de dicho servidor (192.168.1.254), cosa a todas luces inútil cuando trabajamos con tantos equipos y con tanta gente joven, máxime cuando uno de nuestros objetivos es regular el acceso a ciertos sitios como las redes sociales.

Así pues, lo que queremos, no es sólo un proxy. Queremos un proxy *transparente*. Esto es, conseguir que cuando un equipo se conecte a Internet desde nuestra red local, lo haga utilizando como *puerta de enlace* a *granhermano*, en la dirección *192.168.1.254*.

Para hacerlo, necesitamos dar los siguientes pasos:

- Granhermano debe funcionar como servidor DHCP, de modo que cuando los equipos se conecten y busquen de manera automática la IP con la que van a funcionar, sea dicho servidor el que se la de, y se autoasigne como puerta de enlace.
- Debemos configurar *SQUID* como servidor transparente (desde el mismo *WEBMIN* se puede hacer, basta con especificar la opción *transparent* al puerto que viene por defecto en *Opciones de puertos y trabajo en red*).
- Como *SQUID* trabaja con peticiones al puerto 3128, pero las peticiones web siempre se hacen desde el 80, reconfiguraremos *IPTABLES* creando una regla en el servidor (funcionando como puerta de enlace) que desvíe todas las peticiones web al servidor *SQUID*.
- Una vez esté el sistema (más o menos estable) funcionando, podemos crear *reglas de acceso (ACL's)* para determinar en qué

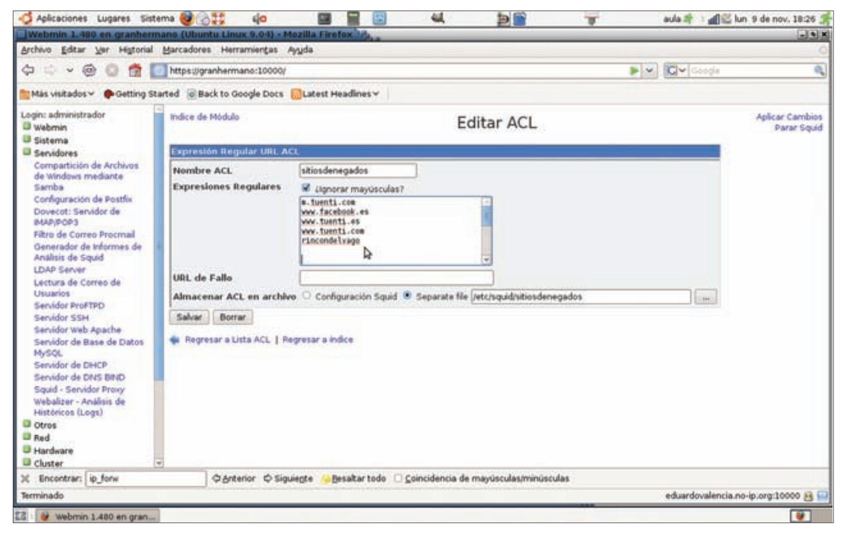


Figura 7. En el ejemplo, definimos las direcciones de Tuenti y Facebook, así como cualquier referencia al rincón del vago

condiciones pueden acceder a qué sitios de Internet cada uno de los equipos.

Configuración del servidor DHCP

El paquete a configurar sería el *DHCPD:* `sudo aptitude install dhcp3-server.`

La configuración del servidor DHCP puede configurarse por *WEBMIN* o de manera manual, en el archivo */etc/dhcp3/dhcpd.conf* (ver Listado 2). En el listado precedente, preparamos a *granhermano* para que conceda direcciones IP desde 192.168.1.50 a 192.168.1.250, reservándonos las otras direcciones para otras funciones (impresoras de red, servidores secundarios, prácticas de creación de redes locales con el alumnado de Bachillerato, etc...).

IPTABLES

IPTABLES es una herramienta cortafuegos que permite a nuestro equipo interceptar y filtrar paquetes de red, de acuerdo a una serie de reglas de una muy sencilla gramática. En su estado original, SQUID trabaja con peticiones de otros equipos a través del puerto (por defecto) 3128, a diferencia de los programas exploradores como Firefox, que realizan sus peticiones al servidor a través del puerto 80. Por ello, es necesario introducir una regla que explique al ordenador que cualquier petición que venga desde la red local por el puerto 80 debe ser redireccionada al puerto 3128 (ver Listado 3).

Estas órdenes, que no deseamos tener que reescribir a cada reinicio del servidor, están en un script que llamamos configurandoiptables.sh y que se ha grabado (siempre como root) en la carpeta /etc/network/if-up.d

A partir de ahora, si todo ha ido bien (lo sabemos, lo sabemos, Murphy es especialmente estricto con los maestros y profesores; es posible que no haya salido a la primera, simplemente, repasemos lo ya hecho y volvamos a intentarlo); si todo ha ido bien, más tarde o más temprano, cada equipo de la red local que arranque a partir de ahora obtendrá su dirección IP de manera automática del servidor DHCP y realizará, sin saberlo, sus consultas TCP/IP a través de nuestro servidor.

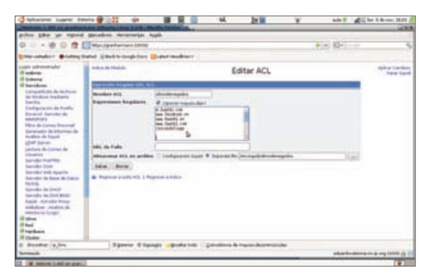


Figura 8. Dando acceso a la acl redlocal exceptuando sitiosdenegados

Reglas de acceso en SQUID. Filtro web

Ya hemos mencionado un par de veces que uno de los mayores atractivos de utilizar *SQUID* en un centro educativo consiste en la posibilidad de sancionar de un modo muy sencillo el acceso a determinadas páginas por parte de nuestros alumnos.Una vez hemos conseguido que todos los navegadores del instituto pasen por nuestro querido *granhermano*, crearemos un par de reglas de control de acceso (*ACL*) muy sencilla a través de *WEBMIN*, la primera (*redlocal*), que definirá el conjunto de ordenadores que tendrá acceso a Internet dentro del instituto, y la segunda (*sitiosdenegados*), marcará el conjunto de páginas web o contenidos que queremos restringir. Una vez conectados al gestor web, en el apartado *Squid-Servidor Proxy*, subapartado *Control de acceso*, podremos hojear las distintas reglas que suelen venir por defecto con esta herramienta. Una de ellas suele ser *localnet*, que define la red 192.168.1.0, máscara de red 24, esto es, abarca todo el rango de direcciones IP 192.168.1.0/192.168.1.255. Podemos modificarla o definir nuestra propia ACL de este tipo, indicándole a *WEBMIN* que queremos definir una regla de tipo *Dirección de cliente.*

La segunda *ACL, sitiosdenegados*, es una regla de tipo *Expresión regular URL.* Podemos indicar direcciones web completas, o palabras relacionadas con los contenidos que queremos restringir. Este conjunto de palabras y direcciones puede guardarse en archivo aparte (es lo que hemos hecho nosotros), o directamente en el archivo de configuración de *SQUID, /etc/squid/squid.conf.*

Una vez definidas ambas reglas, sólo nos falta relacionarlas. En el mismo subapartado de *Control de acceso,* pestaña *Restricciones ICP*, añadiremos la regla *redlocal*, remarcando la excepción *sitiosdenegados*. De este modo, todos los ordenadores de nuestra red tendrán libre acceso, exceptuando las ocasiones en que hagan referencia a los sitios que hemos detallado en nuestra lista negra. Debo reconocer que esta parte de nuestro proyecto fue una de las más complejas, pero también las más satisfactorias. Uno de los mayores temores de muchos compañeros de profesión a la hora de abrirse al uso de las TIC ha estado siempre en el acceso, por parte de alumnos más rápidos que ellos, a sitios web digamos... bueno, más comprometidos. Desde ahora, podemos controlar y restringir el acceso a cualquier tipo de contenido que no sea específicamente educativo, léase diarios deportivos, pornografía, imágenes violentas...

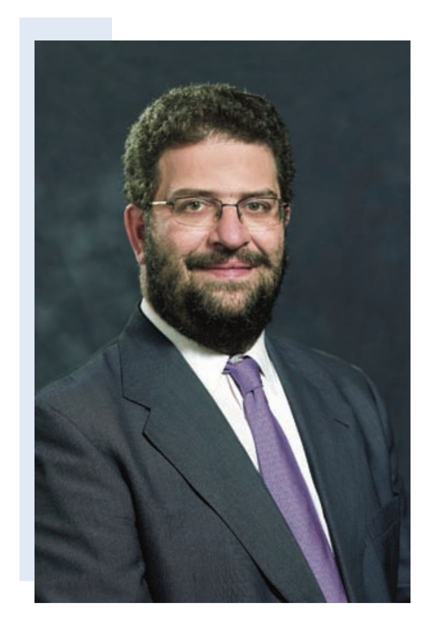
Conclusión

En este primer artículo, hemos emprendido la instalación básica de un ordenador lo suficientemente operativo como para incluirlo en la red local de nuestro centro educativo y permitir que el resto de la comunidad pueda seguir realizando sus actividades sin resultar afectados por ello. Nos hemos asegurado de que nuestra LAN tiene ahora un ordenador central que va a filtrar y controlar las entradas y salidas al exterior de los equipos con los que trabajan nuestros alumnos menores de edad, hemos logrado racionalizar el acceso a Internet, estamos preparados para compartir archivos y carpetas a través de la red de acuerdo a un sistema de permisos racional, y lo más importante, podemos administrar a *granhermano* desde el exterior, de modo que podamos realizar el resto de las tareas de instalación de forma externa; de hecho, en un centro educativo en el que varios grupos tienen que acceder a lo largo del día a las aulas con ordenadores, no podemos depender de tener un momento libre en nuestro horario como profesores, y esperar que justo en ese momento el aula no esté ocupada.

En una segunda parte, nos introduciremos en las apasionantes posibilidades que nos ofrece *APACHE2* para poner a disposición de cada miembro del personal, docente y discente, su propio espacio web (muy interesante para aumentar la cohesión de la comunidad desde el punto de vista de la comunicación), así como el proceso básico a llevar a cabo para instalar un sistema de correo para profesores y Departamentos, a través de *POSTFIX, DOVECOT* y *SQUIRRELMAIL*. También investigaremos en las posibilidades de ampliación de *WEBMIN* a través de los módulos *USERMIN.* ¡Hasta el próximo número! 

Sobre el autor

Ingeniero Técnico Industrial de formación, Antonio Gómez es profesor de Tecnologías en el IES Eduardo Valencia, en Calzada de Calatrava (Ciudad Real), desde el año 2004, donde desempeña el cargo de Responsable de Equipos Informáticos del centro. Anteriormente ha sido también asesor TIC en el Centro de Profesores de Puertollano (Ciudad Real), con el que sigue desarrollando diversos proyectos de innovación y formación relacionados con el uso del Software Libre en educación.



Esclavos
de ordenadores nuevos

Fernando de la Cuadra, director de Educación de Ontinet.com, distribuidor en exclusiva de las soluciones de seguridad de ESET en España

Lo peor que le puede pasar a un profesional de la informática son las Navidades. Reconozcámoslo, una vez pasada la entrega de regalos navideños y de los Reyes Magos, aparecen como salidos de debajo de las piedras los extraños parientes y vecinos que solicitan ayuda para configurar el nuevo y reluciente ordenador que se han comprado aprovechando las ofertas.

Lo primero que debemos tener en cuenta es que no siempre deberíamos encargarnos de estas cosas. Ir a casa del vecino del segundo derecha una sola vez es el equivalente a firmar un contrato de esclavitud permanente. A partir de ese momento nos veremos inmersos en una suerte de tareas en muchos casos desagradables y, cuando menos, incómodas. Recuerdo el ordenador de un amigo al que nunca le pude confesar, para evitar problemas familiares, que su hija, que por lo menos era mayor de edad, era una apasionada de la zoofilia, simplemente echando un vistazo a unos cuantos archivos temporales.

Aproximadamente una vez al mes deberemos volver a esa casa a reparar todos los desaguisados cometidos por los usuarios de ese ordenador: instalaciones defectuosas, virus, secuestros del navegador… y en virtud del contrato ficticio de esclavitud, no podremos abandonar la casa hasta que no esté arreglado.

La tentación sublime que hemos tenido todos es formatear el sistema e instalar alguna distribución Linux que sea sencilla de manejar y que no nos de problemas. Grave error. Los usuarios de ese ordenador no están demandando un sistema operativo determinado, ni una serie de aplicaciones en concreto. No es que quieran chatear: quieren el Messenger, con sus iconitos tan simpáticos y su publicidad de apuestas y casinos. Y no es que quieran utilizar un procesador de textos: necesitan Word. Da igual que no tengan licencia, da igual que los haya gratuitos más rápidos, ligeros y seguros, quieren prolongar su oficina en casa.

Además, la instalación de un sistema Linux va a hacer que no puedan romper a gusto el sistema. Si de repente no se puede estropear el arranque del sistema borrando unas cuantas claves de registro, esa cosa llamada Linux no merecerá la pena.

Antes de aceptar cruzar la puerta de ese vecino o familiar, debemos pensar seriamente a qué nos exponemos, y en lugar de entrar en esa casa con unos cuantos CD o DVD, debemos informarnos de cuántos centros de formación informática hay cerca de su casa, y hacérselo saber. Los ayuntamientos suelen tener muchísimos cursos que, encima, son gratuitos, y pueden ayudar no solo a manejar un ordenador decentemente, sino a hacerlo sin ayuda de terceros y de manera segura.

En el peor de los casos, aunque no podamos evitar ser el centro de soporte técnico de la escalera, sí que conseguiremos cambiar la manera de preguntar las cosas. Es increíble cómo cambia la frase "es que no sé que le pasa al ordenador" a "me he instalado un acelerador de descargas y no me funciona el navegador". Habrá que acudir de todos modos ("hombre, acércate que son amigos de toda la vida" es una frase demoledora en nuestro entorno doméstico), pero nos acabamos de ahorrar una hora de descifrar frases como "el dibujito de escribir no está donde siempre".

Y lo peor de todo: al igual que todos los años, seguiremos recordando la frasecita de marras: "Manolete, si no sabes torear… ¿para qué te metes?"

Páginas recomendadas



www.diariolinux.com



www.elguille.info



www.gatolinux.blogspot.com



www.opensourcespot.org



www.hispabyte.net



www.linuxdata.com.ar



www.linuxhispano.net



www.pillateunlinux.wordpress.com



www.usla.org.ar



www.mundopc.net



www.picandocodigo.net



www.linuxuruguay.org



CONCURSO UNIVERSITARIO DE SOFTWARE LIBRE





Las últimas noticias al instante sobre todos los proyectos

PATROCINA



ORGANIZA





COLABORADOR PRINCIPAL





MEDIOS OFICIALES







COLABORA

















